FACULTY OF TECHNOLOGY

Real-time data quality monitoring and improvement in energy networks

Henri Välikangas

PROCESS ENGINEERING

Master's thesis

September 2021

ABSTRACT

FOR THESIS University of Oulu Faculty of Technology

Degree Programme (Bachelor's Thesis, Master's Thesis) Process Engineering		Major Subject (Licentiate Thesis)	
Author Henri Välikangas		Thesis Supervisor Mika Ruusunen, D.Sc. (Tech); Jari Ruuska, D.Sc. (Tech); Ari Vuokila, D.Sc (Tech); Petri Hietaharju, M.Sc (Tech)	
Title of Thesis Real-time data quality monitoring and improvement in energy networks			
Major Subject Automation technology	Type of Thesis Master's thesis	Submission Date September, 2021	Number of Pages 99

Abstract

Data quality monitoring is an important aspect in real-time data-based operation and of growing interest. Studying the different methods and approaches in real-time data quality monitoring, in the context of the energy systems, can yield some highly beneficial improvements in the ever-growing demand for material efficiency and energy savings. Quality flags, based on appropriate quality dimensions, can improve the decision making of systems in real time. The goal of this study is to find out, how this can be applied, utilizing the varied and large volumes of energy industry data.

The concept of data quality was first dissected at a theoretical level, to understand what meaningful data quality dimensions in the energy systems could be, in terms of possible sources of data and what aspects of it are meaningful for the quality of the processes. Based on the gathered understanding from the related theoretical section, an understanding of essential data quality dimensions was formed, helping in the choice of data quality dimensions for this study. After this, the potential data quality pre-processing and analyzing methods were inspected. The goal was to apply simple methods to see what results could be achieved with them when the data quality flagging algorithm was formed. Selected seven quality dimensions were Accessibility, Interpretability, Completeness, Consistency, Timeliness, Accuracy and Believability. Data was generated with imputed errors, and the data quality flagging algorithm performance was tested on it, simulating three signals producing sensor readings, one with redundant readings, two without.

The data flagging results were correct in all simulated cases, but the accuracy of the estimated values varied. High precision data quality description about the data compared to the actual value was achieved consistently with the signals that had redundant values utilizing the chosen simple methods. On the other hand, algorithm produced less accurate estimation value with the signals without the redundant readings, depending on the error type. Drifting error type was challenging to handle if only one signal was available, without more sophisticated estimation methods.

Most data quality checks studied in this thesis are applicable in real time operation, but changes are needed in the estimation methods for the individual signals. The selected methods were simple to ease the load on real-time data quality monitoring requirements. Further research should concentrate in finding better methods to deal with the errors that caused a lot of estimation challenges in this study.

Keywords: Data quality, Quality dimensions, Real-time operation

<Additional Information

TIIVISTELMÄ

OPINNÄYTETYÖSTÄ Oulun yliopisto Teknillinen tiedekunta

Koulutusohjelma (kandidaatintyö, diplomityö) Prosessitekniikka		Pääaineopintojen ala (lisensiaatintyö)	
Tekijä Henri Välikangas		Työn ohjaaja yliopistolla Mika Ruusunen, TkT; Jari Ruuska, TkT; Ari Vuokila, TkT; Petri Hietaharju, DI	
Työn nimi Datan reaaliaikainen laadunhallinta energiaverkoissa			
Opintosuunta Automaatiotekniikka	Työn laji Diplomityö	Aika Syyskuu, 2021	Sivumäärä 99

Tiivistelmä

Datan laadun varmistaminen on tärkeä osa sen reaaliaikaisessa hyödyntämisessä ja kasvavan kiinnostuksen kohde. Energiateollisuuden kontekstissa datan laadun reaaliaikaisten monitorointimenetelmien tutkiminen voi tuottaa hyödyllisiä tuloksia tehokkuusvaatimusten jatkuvan tarpeen kasvaessa. Dataa hyödyntävien järjestelmien päätöksentekoa voidaan parantaa reaaliaikaisella laatuliputuksella, joka kertoo käsiteltävän datan laadun sidottuna sen tärkeisiin laatudimensioihin. Tämän tutkimuksen tavoite oli selvittää, miten tämä voidaan toteuttaa monimuotoisella ja runsaslukuisella energiajärjestelmien datalla.

Työ alkoi datan laadun määrityksestä perustasolla, että ymmärrys datan laadusta energiateollisuuden kontekstissa voitiin muodostaa. Tähän liittyi datan laatudimensioiden tunnistaminen ja niiden soveltaminen energiajärjestelmissä. Valittaviin laatudimensioihin vaikuttavat datan alkuperä, sen määrä ja tyyppi. Tämän jälkeen arvioitiin mahdollisia esikäsittely ja analyysimenetelmiä datan laadun valvonnan kannalta, kehitettävää reaaliaikaista algoritmia varten. Seitsemän datan laatudimensiota, joita tässä työssä käytettiin algoritmin määrityksessä, olivat esteettömyys, tulkittavuus, täydellisyys, johdonmukaisuus, ajallisuus, tarkkuus ja uskottavuus. Kehitettyä algoritmia testattiin simuloidulla datalla, johon oli lisätty virhettä tietyille aikaväleille ja satunnaisia virheitä. Simuloituja signaaleja oli kolme, joista yhdessä oli redundantteja datajoukkoja.

Simulointitulosten perusteella datan liputusarvot olivat oikein kaikissa tilanteissa, toisaalta estimaattien tarkkuus hetkellisestä arvosta vaihteli. Korkea selitystarkkuus datan hetkellisestä laadusta verrattuna datan oikeaan arvoon saavutettiin johdonmukaisesti signaaleissa, missä oli redundantteja mittausarvoja ja kun sovellettiin yksinkertaisia menetelmiä. Signaalien ryömintävirhe aiheutti haasteita yksittäisiin mittausarvoihin perustuvilla estimaattoreilla, joka viittaa kehittyneemmän estimointimenetelmän tarpeesta tulevaisuuden tutkimuksen kannalta.

Tulosten perusteella suurin osa työssä testatuista datan laatutarkastuksista soveltuvat reaaliaikaiseen monitorointiin, mutta estimaattien tarkkuuden parannus vaatii muutoksia estimaattimetodeihin etenkin, jos saatavilla on vain yksi mittausarvo. Yksinkertaisten menetelmien valinnan syy oli helpottaa reaaliaikaisen laatuliputuksen asettamia vaatimuksia datan laadun monitoroinnissa. Jatkotutkimus puuttuvien ja virheellisten arvojen estimaattien parantamiseen on tärkeää.

Asiasanat: datan laatu, laatudimensiot, reaaliaikaisuus

Muita tietoja

PREFACE

The goal of this thesis was to research data in energy networks as a part of project HOPE, as a masters degree programme at the University of Oulu. The objective was to do research into the energy network data, and how to manage it efficiently with ever increasing volumes of data coming from advancing technologies. Optimal solutions were to be studied to solve these challenges.

The first subject to do I was hired in December 2020 to start doing this thesis as a research at the University of Oulu and it continued until September. I got all the support I wanted, but I rarely knew better to ask. I got to work with great people and the research was a challenge.

I thank Mika Ruusunen, Jari Ruuska, Ari Vuokila and Petri Hietaharju for guiding me through my thesis. Thanks for your patience. I also thank the University of Oulu for letting me get this job.

Oulu, 17.9.2021

Henri Välikangas

Henri Välikangas

TABLE OF CONTENTS

1 INTRODUCTION

Data quality is becoming more of a focus, as the volume and the heterogeneous nature of data is increasing, more information is available, and data is relied upon more. This is especially the case in the energy field, where real-time adaptability of the systems is increasingly required. To achieve real-time adaptability a lot of high-quality data is needed. Ranking the incoming data based on quality gives the system data options to base the decisions on the incoming data to achieve this goal.

Data quality can be broken down into quality dimensions, which address different aspects of the of the data. By doing this, the quality can be assessed more thoroughly, and the possible source of bad quality data can be more easily recognized. This is especially useful in energy networks, where bad data quality can cause system wide problems. This is true in the customer, production, and transmission side of the network.

Real-time assessment of data quality can be achieved by quality flagging the incoming datapoints to the energy network, which helps to achieve the previously stated requirements for the data quality. The quality information of a datapoint, found in the quality flag, can be utilized in the system giving less weight to the poor-quality data and focusing weight on the high-quality datapoints. This leads the system operating as intended, minimizing the ill effects caused by bad quality of data.

The goal is to understand data quality in general and then apply that in practice. This will be achieved by going over what data quality means in theory, and then to understand how it is divided into different data quality dimensions. These data quality dimensions will be the tools in the practical data quality monitoring simulation, where the data is flagged, or rated based on its goodness in each of the chosen quality dimensions.

The challenge comes from on-line approach to quality of data. It is a restriction as it demands quick and accurate results. This mandates for choosing methods that are easy to apply, and to specify the quality dimensions specifically in the framework of energy system in the case of this study. The goal is also to build a quality monitoring system, that could someday operate in real-world applications.

2 DATA AND ENERGY NETWORKS

The energy networks have data coming in from many sources. This needs to be addressed by branding the data according to its quality. This means utilizing methods to assess the data quality through quality dimensions that are present in all data. This helps with the decision making in the energy network and is a critical part of the optimization process in the energy network operation.

Data quality is tied to the data quality dimensions, as they describe the data quality in qualitative and quantitative descriptions of real-world characteristics and numeric values. Thus, the data quality control and optimization are done by satisfying the conditions defined by the data quality dimensions in context to the optimization or control target. Data quality has set requirements depending on the application. Energy networks are the target under investigation in this study and they set some obstacles to overcome, such as analysing the sheer volume of data. It is called “Big data analysis” and it sets the data quality dimensions that allow for handling high volume and variety at a higher standard than the other data quality dimensions.

The continuous data stream quality control may set limits on what methods can be used, as the volume of data is vast, and the monitoring and control actions must happen swiftly. Handing and analysing this big data requires methods that can handle heterogenous and big volume data. Also, energy networks set real-time requirements, that need to be satisfied. Purpose of high-quality data is to make sure operation is based on valid and accurate information.

2.1 Introduction to energy networks

Energy network is the main concept under which all energy related network structures fall. Energy network itself consists of energy conversion, -transfer and -distribution (Rismanchi 2017). Different types of energies have different kinds of subnetworks, for example electrical, thermal, and fuel networks, which in turn means that they have their own grid infrastructure, and produce their own type of data. (Diamantoulakis *et al*. 2015)

The energy network generates data throughout its structure, which is collected to maintain intended operation condition throughout the network. Generation, transmission, and consumer side can be identified in the network. The origin and type of collected data thus becomes important factor, in the heterogenous and real-time data flow. Validating measurements becomes important, as mistakes happen in the system, and the data is relied upon. The data sources and how to evaluate their quality becomes a problem to solve. (Allalouf *et al*. 2014)

Smaller sub-networks, that operate individually as a part of a bigger network, make up the energy network. Subnetworks optimize their own operation based on the data that they produce locally, which can be further utilized in the whole energy network optimization. The data collected from the subnetworks can be pre-processed and filtered to begin with, which helps with the volume of the data, but might leave out important information. (Cheng *et al*. 2018)

2.2 Energy network structure

Energy network is composed of energy supply subnetworks of different forms of energy, that all produce data and receive data (see Figure 1). These different forms of energy are coupled together, which means transmission and conversions of energy between the subnetworks. The subnetworks are thus distributed parts, that together make up the whole energy network. The operation is optimized on the energy network level. (Chen *et al*. 2017; Chen *et al*. 2019)

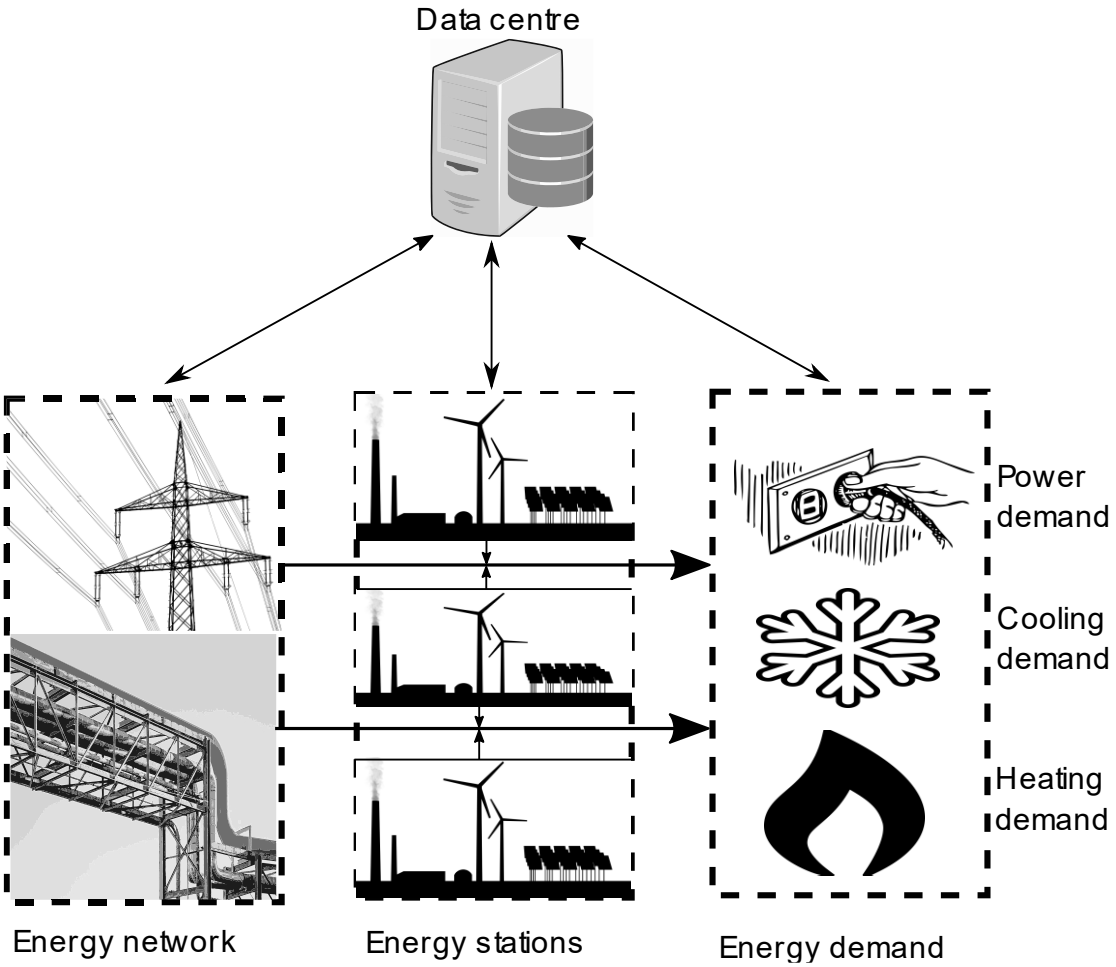

Figure 1. Energy network structure, adapted from Xie *et al*. (2020) and Xu *et al*. (2017).

The subnetworks generally consist of thermal, electric and gas networks, with locally integrated renewable energy sources and storage capabilities (Li *et al*. 2018). These different forms of energies are utilized together to satisfy the needs set upon the whole network. The information from all sources in the networks are traditionally gathered into the data centre. The information flow from the subnetworks to the data centre allows for generation, transmission, distribution and operation control, as seen in Figure 2. (Xu *et al*. 2011)

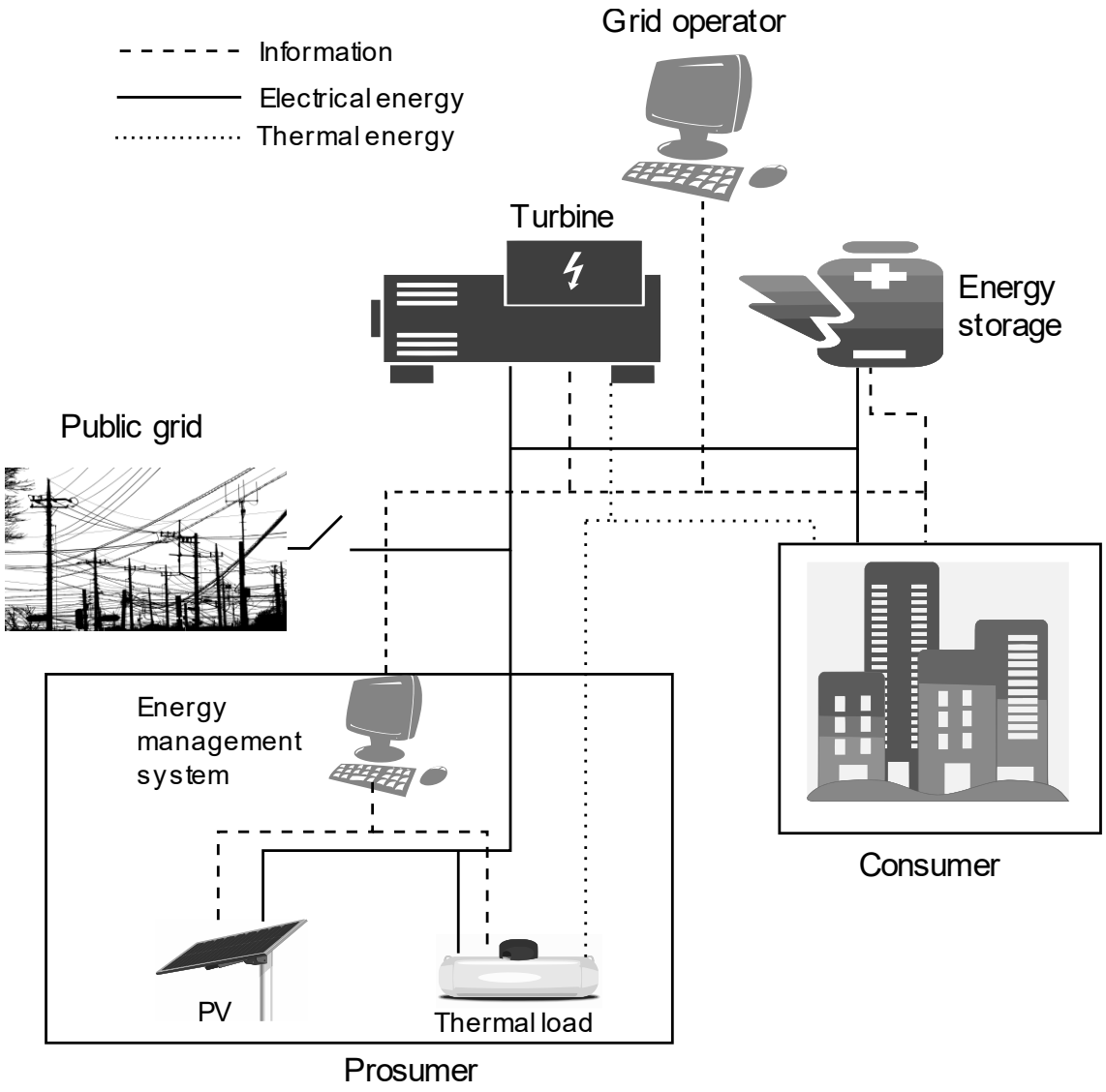

Figure 2. Microgrid energy network structure, adapted from Liu *et al*. (2018).

Traditionally the communication network exists between the main station and the substations. The information flows into centralized main station from the substations, which is then utilized to control the whole network operation, through automation system, as seen in Figure 3. More recently, with the advent of smart two-way communication solutions, the structure requirement of the communication network has become more complex. This has led to distributed control and command structures, that are making more accurate control decisions locally, while still operating as a part of the energy network. (Ancillotti *et al*. 2013)

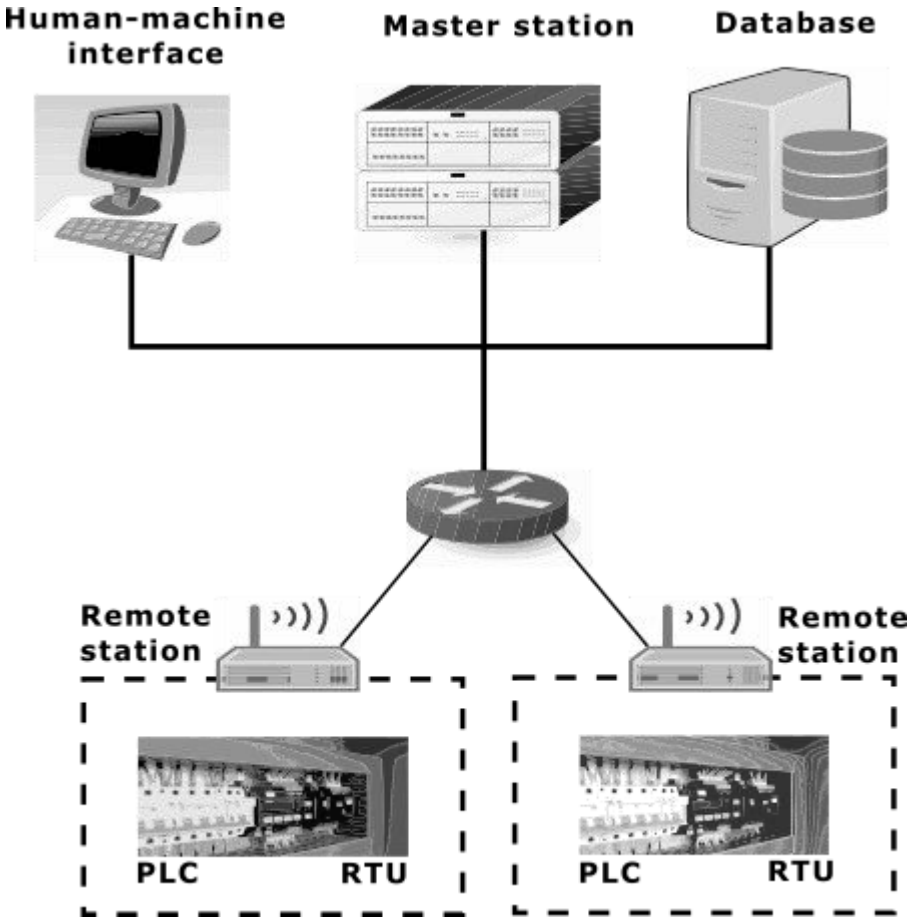

Figure 3. Communication network structure, adapted from (Ancillotti *et al*. 2013).

Multi-energy network structure is suggested by Li *et al*. (2018) in their study. The goal of the network structure is to be able to answer to as many energy forms demands as possible. Energy forms such as various fuels, e.g., natural gas, hydrogen, electricity etc., and different forms of direct energy to end users. These different energy forms can interact with each other, and the energy flows can thus be handled more efficiently, balancing the load between them optimally. Combined heat and power (CHP) is a good example of this, as it can utilize many fuels and can produce both heat and electricity.

A regional integrated energy system (RIES) is a regional energy supply network formed by the coupling energy systems such as electricity, natural gas and thermal (cold) energy. There is little information available about multi-energy networks, where the multi-energy flows can access the grid, heat networks and natural gas networks in the literature. At this moment, the distinct energy sources act largely as individual parts, not

as one, which can make the energy network optimization more difficult. Thermal energy is suggested as a buffer and the media for other energy sources, in these multi-energy networks, turning other energy sources into heat and vice versa. The data utilization of thermal energy is beneficial in improving the whole network operation, which puts more emphasis on thermal network operation and distribution as a whole. (Tang *et al*. 2018)

2.3 Data sources in energy networks

Energy networks are composed out of individual networks of different energy forms and sources, that together form an energy network. Different individual sources generate individual data, which increases the overall data amount, as seen in Figure 4 (Tu *et al*. 2017). The energy network is in between the consumer and the production of energy, and encompasses data from both, the production and consumption. In addition, there is energy transfer data (infrastructural data) and external data (third party, like economic, weather and meteorological data) that affects the optimal energy production and transfer conditions in the energy network (Gürcan and Yazici 2017).

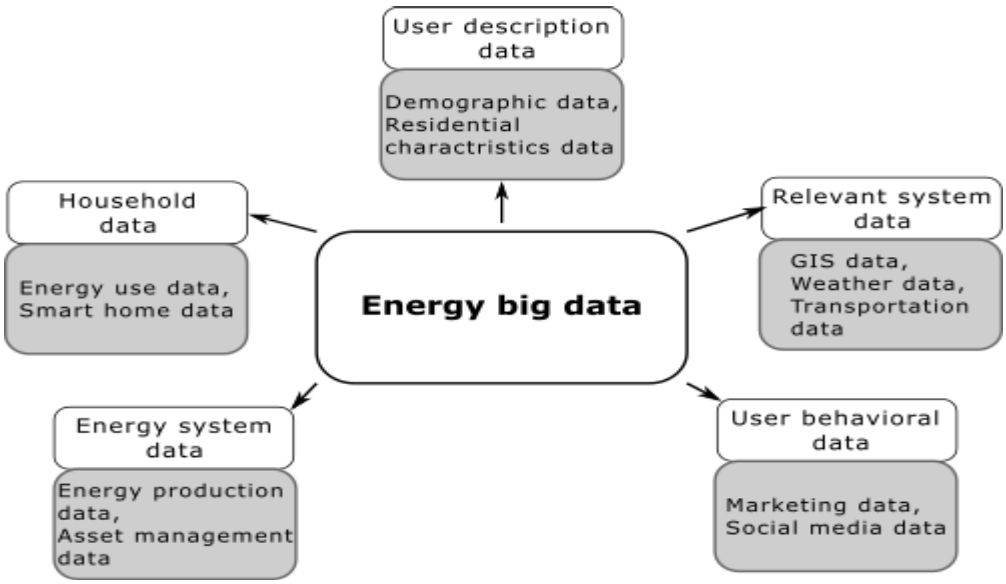

Figure 4. Big energy data, adapted from Zhou and Yang (2018).

The data in the energy network operation is captured by a multitude of different types of sensors, depending on the source and type of data. The grid level (transmission) on-line

operation data is captured by phasor measurement units (PMUs), smart meters on the consumer side and various sensors and actuators on the production and third-party side (Radhakrishnan and Das 2018). The PMU data is about voltage magnitude, and frequency and phase angle, which gives information about the frequency difference caused by differences in production and consumption (Shalalfeh *et al*. 2020).

There are various sensors distributed within the energy network. Their function is to monitor and send information about the systems and devices they monitor to make control of the whole energy system possible (Jaradat *et al*. 2015). In addition to the traditional sensors monitoring what happens inside the network, external factors like weather data, market data, and geographical data can be utilized to improve energy network operation (Alahakoon and Yu 2016). Consumer side data can be collected through smart meters (Zhou and Yang 2016).

The advent of big data has led to the increase in data volume and heterogeneity in content and in structure (see Figure 5). This has made the traditional database mechanisms insufficient to handle the big data. The processing, storage and analysis requirements need to adapt to handle big data, especially in energy industry, where huge amounts of data is produced and handled. (Gürcan and Yazici 2017)

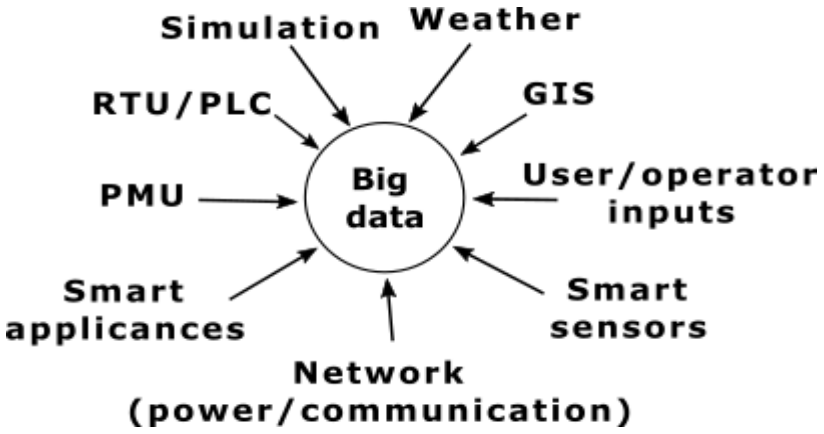

Figure 5. Data from multiple sources adapted from Bhattarai *et al*. (2019).

Data quality becomes relevant because it affects the whole energy network. The goal is to not lose any of the information in all of the available heterogeneous (big) data, and to

focus on the relevant information within said data to reach optimal control over the energy network (Rusitschka and Curry 2016). There is a lot that can go wrong with this much data variety and volume, which leads to a critical need to focus on the data analytics and quality control in order to manage the network operation (Hou *et al*. 2019). The key problems in the energy big data are data volume, uncertainty, security and time synchronisation (Bhattarai *et al*. 2019).

3 DATA QUALITY

The first step towards evaluating data quality is to determine what quality means in context of data. The actions done upon the data need to be explained in an understandable manner to make it comprehensible to the man on the street. The next step is to fit this definition of data and its dimensions into energy network data, in a way that reasonable and valuable results can be gained.

Firstly, it is important to differentiate between information and data as these terms are often used interchangeably. The difference lies in the processing state of these two terms. Data is used to mean raw data from the system and information is used to mean processed data, where the information within the data is found and understood or utilized. (Wang *et al*. 2001)

Wang and Strong (1996) define data quality as "*data that are fit for use by data consumers*" and data quality dimensions as "*a set of data quality attributes that represent a single aspect or construct of data quality"*. Consumer in this case are the data consuming processes within energy networks. These definitions give a good framework for assessing data quality and how it needs to be done in context to the applied area of interest. Context is very important in assessing data quality, as the data quality indicators as in data quality metrics need to be based on the context area, or they will be of no use, for example, blue is not a good indicator to electricity grid load.

Data quality problems are defined by Strong *et al.* (1997) as issues encountered in one or more utilized data quality dimensions, that render the data mostly or completely unfit for use. The intrinsic value of the data can be lowered by the data source, due to differences in data between sources, which leads to lowered believability which then leads to data not being used (see Figure 6). Good quality data is thus data that fits the intended purposes and holds the needed relevant information. High quality data relates to more than just good data quality metric specifications, as data might be used in many different tasks. This necessitates a good overall data quality over all the data quality dimensions.

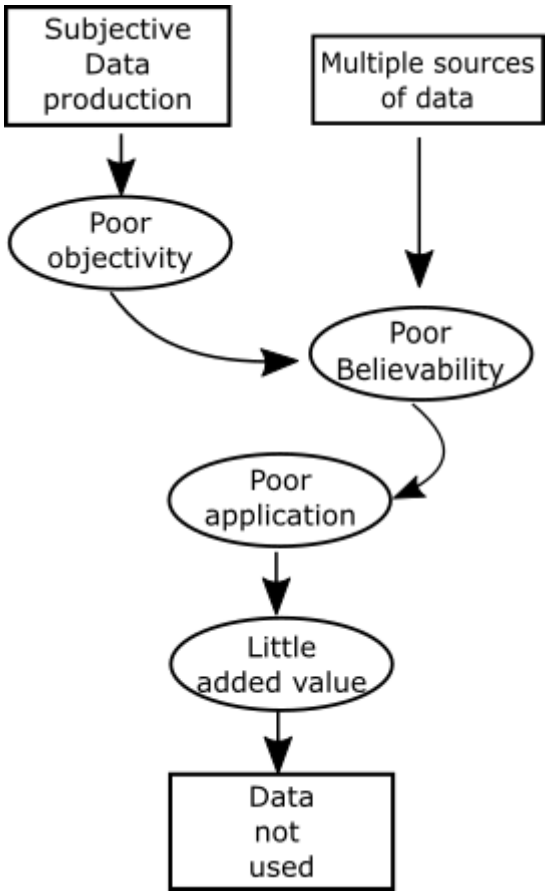

Figure 6. Sources of data quality problems, adapted from Strong *et al*. (1997).

3.1 Implications of quality

Data is a representation of specific characteristics of objects, events, and concepts. In a sense data is a model of reality (Sebastian-Coleman 2013). If this presentation of reality is flawed in some sense, issues arise. If operation is based upon reality, it cannot go well if the reality is warped or distorted in some manner. Raw data itself is in a form that cannot be efficiently utilized. That is why the raw data need to be converted/processed into a form that is more suitable for the system, highlighting the informative parts within the data (Phan and Chen 2017).

Data quality is detrimental to any system that utilizes, produces, or receives data in one way or another. The implications of this are quite severe both in the positive and the negative sense. Good data quality can lead to high level functioning and optimization of the process, while low quality data can lead to several problems depending on the data quality flaws. (Wand and Wang 1996)

Poor data quality is a detriment or even a risk when operation is based upon data and the information it brings into the operation. The risks range from monetary to operational inefficiencies, as poor data quality can hamper the operation regardless of the system state and optimization. These factors impose great potential rewards for the study and advancements of the data quality control field. (Liu *et al*. 2020)

Organizations of all kinds are trying to get better value out of the current data they are utilizing. Without investment and attention directed to data quality control and maintenance, the procedures to improve value from data are not necessarily advantageous. Bad data quality can be a costly risk factor to the operation of any type of organization. (Olson 2003)

Big data can be mined to reveal valuable information from the continuous data stream. This can then be utilized to increase the overall information, insights and new ideas gained from the data. This leads to the ability to learn more from already established systems and abilities to improve them just based on the information gained from the data, that was not available before the big data era. (Mayer-Schönberger and Cukier 2013, p. 117)

The data quality assessment reliability is dependent on the reliability of the information under analysis (Ardagna *et al*. 2018). Data quality metrics, based on data quality dimensions, give indication about the data quality in context to inspected data and its origin. "*Objective data quality metrics, like invalid values or missing values, aren't necessarily tied to the performance of the system*", which raises questions about the impacts of specific data quality metrics and thus warrants more inspection towards what is valid data in relation to performance (Loshin 2011a). Data quality can be assessed both subjectively and objectively. The goal, in the end, is to reach an objective assessment, that can be used in practice for the data quality problem at hand. (Pipino *et al*. 2002)

Results and effects of poor data quality can be seen and experienced all the time without necessarily knowing the source of it or drawing the connection between a bad result and bad data quality. Errors and mistakes caused by bad data quality can often be blamed on the system, when in fact the system is not at fault. The system itself would function

completely fine if data quality were at the level the system was designed to function on. This indicates the need to integrate information across different sources of data to fix these problems. (Batini 2016, p. 2.)

3.2 Possible sources of errors in data

Where does the inaccurate data come from, is something one must understand before data accuracy can be fully understood and mastered and why programs to monitor, assess and improve the data quality are required. All the possible sources of error need to be considered in the quality control to reach accurate data. Many sources are listed in many parts of the data handling process. Four general sources of error can be identified as seen in Figure 7. (Olson 2003, p. 43)

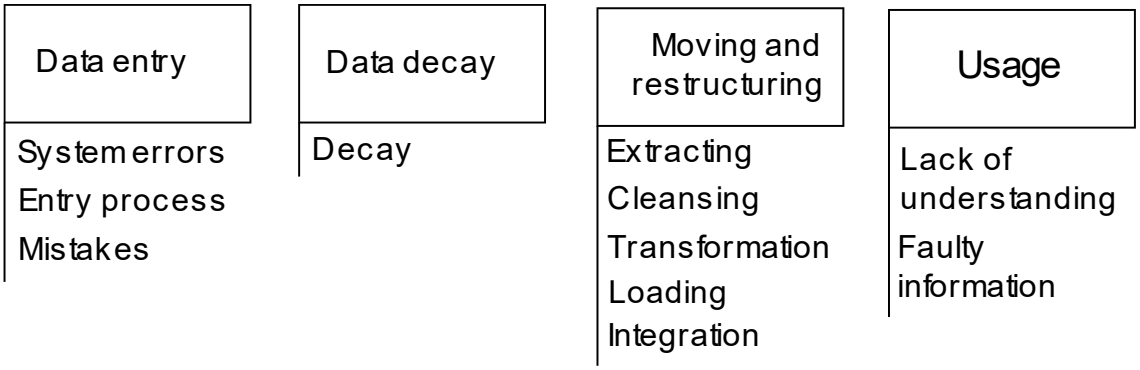

Figure 7. General sources of error in data handling process, adapted from Olson (2003).

Data entry mistakes are the most common source of inaccuracy. The problem may stem from unintentional mistakes or from flawed data entry setup in the system. The system may be blamed for the error, but often, the system works as intended, and the system related errors are due to external actions. For example, if colours are entered into the system and instead of red it comes as read, or bleu instead of blue. The problem might also be in the value, that is entered/registered wrongly from the sensor. (Olson 2003, pp. 44–49)

If no mistakes or errors occur, data should be an accurate representation of the real-world values. The data value does not change, if not changed in the system for a reason

after the initial giving of the data value. So, the data value does not change, but the accuracy does (see Figure 8). The metadata describing the data content is an indicator of decay, as it usually contains information of the data source and time of its creation. This ties together with the timeliness data quality dimension. Timeliness is thus correlated with accuracy. Some data decays, and some does not. Object data, for example, does not decay and the information about data decay should be included in the metadata. The decay of data encourages the need for corrective actions. (Olson 2003, pp. 50–51)

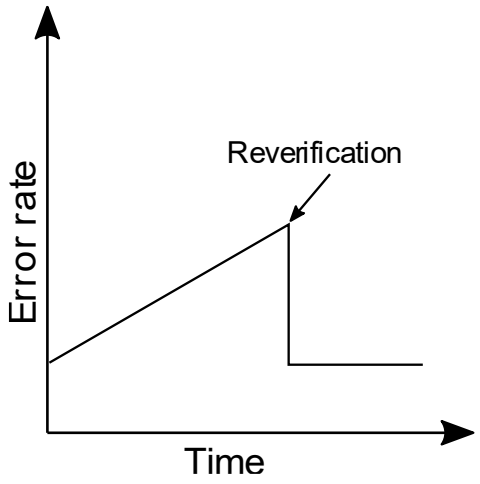

Figure 8. Data decay, adapted from Olson (2003).

The inaccuracies are often created into perfectly accurate data by moving and restructuring it within the framework of the system. By collecting data from data flows into purposes outside the actual data usage, the data is required to be extracted, transformed, and loaded, with a possible data scrubbing process involved. These are sources of inaccuracy, because when the original raw data changes form to fit the systems framework, there is more possibility that the original information inside the data is altered. (Olson 2003, pp. 52–59)

Data cleansing deals with invalid values. Incorrect data values extracted from the source are identified and then the data values are corrected or rejected. Problems arise also from rejecting values that could easily be corrected within the system. The data cleansing corrections are done following a predetermined set of actions and value ranges fitting the process, which can lead to important information being discarded and

the structure of the data changing, leading into structural problems. Figure 9 illustrates the possible errors induced to the data in the steps with hollow arrowheads. (Olson 2003, pp. 59–60)

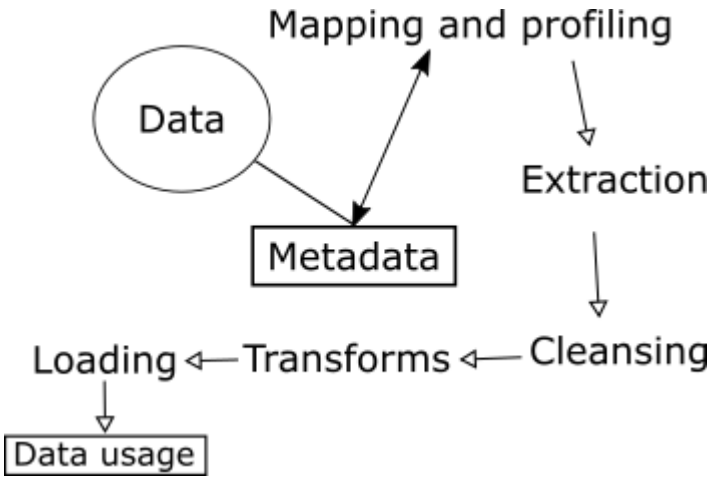

Figure 9. Possible errors sources when moving data, adapted from Olson (2003).

The last place where data inaccuracies occur, is the actual usage of data. It may be as accurate as it can be, but if it is misunderstood and used wrongly in the system, the accuracy does not matter. This is an issue with metadata. Quality levels attached to data through data flagging are a way to make the efficient usage of data easier for the system and the users of said data. (Olson 2003, pp. 62–63)

The scope of problems is quite large. Errors to data accumulate through the whole process where data is involved in. Creation of data, through decay, data transfer (extraction and loading), and use are all possible sources of data inaccuracies. The best way to combat the problems with data quality, is to build a good framework around a specific area where the data is used, and the data is created, transferred, and utilized only there. Data transfer should be linear from the source to the usage, so possibility for error is minimized. Good understanding of the system and variable factors within said system, where data is generated and used, should be achieved. (Olson 2003, pp. 63–64)

The amount of data, to process and handle, has been increasing throughout the years. As technology improves, more data is available for utilization and better decision making by machines and people alike. But if the data quality is low, the added benefit from more data suffers and value is lost. What this means for data quality control, is that the increased amount of data will lead to increased number of inefficiencies and potential losses in the operation if data quality is not sufficient. Therefore, high-quality data, that meets the expectations made to it, is needed. (Sebastian-Coleman 2013)

3.3 Cumulative error

Olson (2003, pp. 8), states that a single wrong value rarely has any significant effects, but cumulative errors from many sources can easily snowball out of control. Errors that go unnoticed and unfixed will take the process one step closer to the inaccuracy tolerance threshold, one at a time, lowering the overall data quality. The error sources might not even be connected in any way or form, but the cumulative effect still exists.

It must also be mentioned, that a generated erroneous datapoint will very rarely be the last one of its kind. What this means in practice is that inaccurate data will not fix itself, almost always continue being inaccurate until the root cause for the inaccuracy is fixed. (Olson 2003, p. 8). The error might be in a form of duplicate, inconsistent, missing or outlier value, and error will persist, if the data quality control method e.g., quantitative data cleansing (QDC), cannot handle or detect the error (Tayi and Ballou 1998).

According to Wang and Strong (1996), data accuracy and objectivity alone are not a sign of high-quality data. There are also other factors besides the mentioned ones, that have great impact on the data quality, and if unaddressed, can lead to drastic problems. Data quality thus needs to be observed from multiple angles, on application basis.

3.4 Quality dimensions

When talking about data quality, it is important to set some definitions upon the perceived data to analyze its level of quality from different viewpoints. This gives the ability to assess the data more thoroughly, get more information about the quality of certain data and to define the requirements of high-quality data. Better data quality

control can be made with more information, such as qualitative dimension information. (Wang and Strong 1996)

Data quality dimensions can be divided into four defined categories: Intrinsic, contextual, representational and accessibility. The dimensions were classified into these categories by utilizing preliminary information on the subject and gained information on data quality requirements from data consumers through surveys. The dimensions were created in context to the source of survey answers, so the result would be a generalized view on data quality dimensions. The data quality categories and the dimensions within them can be seen in Figure 10. (Wang and Strong 1996)

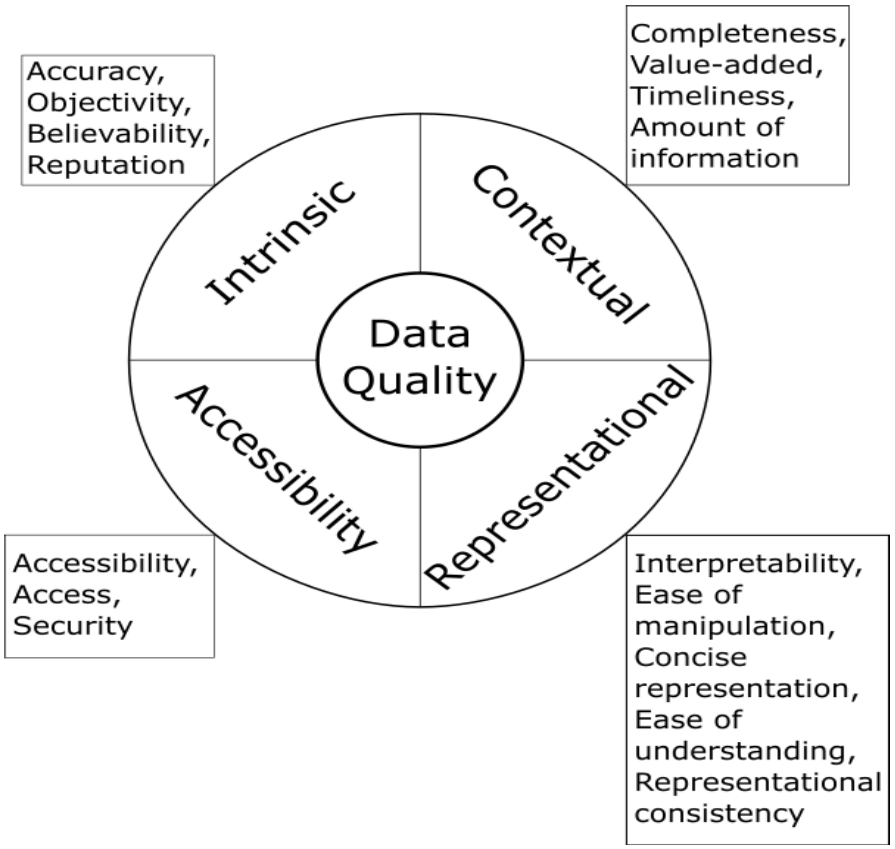

Figure 10. Common data quality dimension categories adapted from Strong *et al*. (1997).

Intrinsic data quality is defined as the data having quality in its own right and relates to the actual values in the data in context to the application. Contextual data quality focuses on the validity and consistency data quality in context to the task.

Representational and accessibility data qualities focus on the computer system side, that stores and provides access to data and information. The system must present the data in a form that is easy to understand and manage concisely, consistently and in an accessible, but secure fashion. (Lee *et al*. 2002; Loshin 2011b)

The data quality dimensions, need to be assessed and combined in a way, that produces a simple, yet effective, combination that can easily be transformed into a usable form. Form that can then efficiently and categorically control and ensure the data quality to be sufficient for the operation, without big trade-offs between used quality dimensions (Batini 2016, pp. 44–45). These kinds of goals for data quality cannot be achieved without having a proper understanding of the data quality dimensions (Tayi and Ballou 1998).

Specific data quality problems need to be addressed with methods, that utilize the data quality dimensions identified to be most important in energy network context. There are seven such dimensions out of the ones listed before in Figure 10, including the categories they belong to in the framework. These are listed in Figure 11, and they are also connected to the data quality problem they address in energy networks. Some relationships between the quality dimensions can be also seen in Figure 11. (Ge *et al*. 2019)

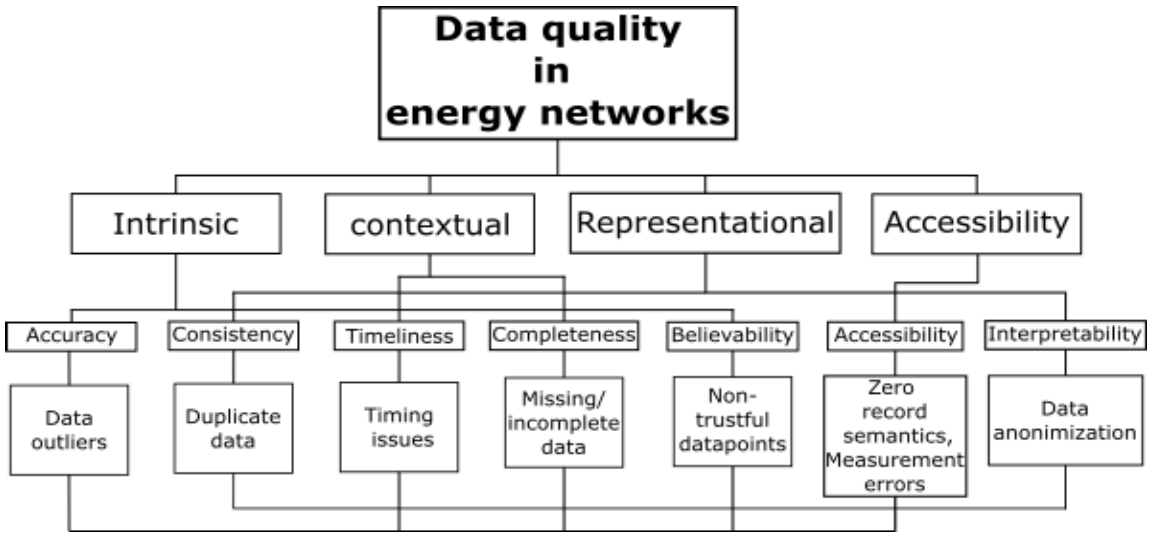

Figure 11. Data quality in the framework of smart grids, adapted from Ge et al. (2019).

3.5 Big data

Information for which traditional methods of processing or analyzing are not sufficient, is defined as big data. The characteristics of big data are defined as three V's, volume, variety, and velocity (see Figure 12). These characteristics define the challenges and opportunities presented in big data and further in big data processing and analysis. (Zikopolous *et al*. 2012, pp. 3–5)

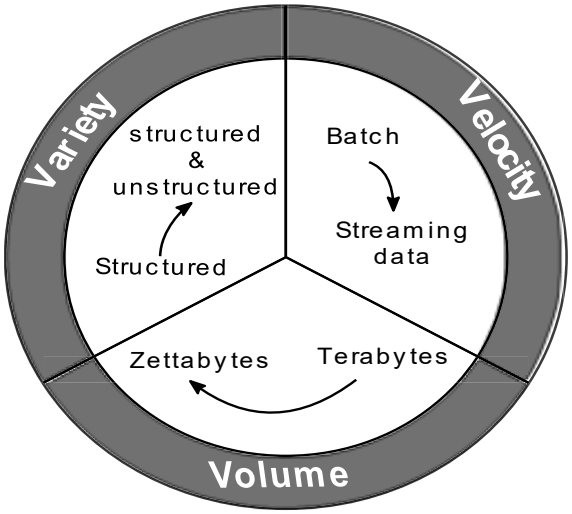

Figure 12. Three V's of big data, adapted from Zikopolous *et al*. (2012)

Volume implies the amount of data; more specifically big data is defined ranging from terabytes to petabytes and eventually to zettabytes. Variety means the different types and structures in data. The big data contains raw, unstructured, and semi-structured data from different sources, increasing the overall complexity of data in general. Velocity implies the speed by which data is created and how fast it needs to be handled (Zikopolous *et al*. 2012, pp. 5–9). Two additional V's can be added, value as in extracting the hidden significant information out of the big data (Mayer-Schönberger and Cukier 2013, pp. 94–97), and veracity that refers to the accuracy or correctness of information from the source (Bello-Orgaz *et al*. 2016).

The data quality dimensions between traditional data and big data are explained by Ramasamy and Chowdhury (2020), and the basis of the quality is the same in both. The difference comes with the increased volume, variety, and even real-time stream of big

data, which renders dealing with every individual datapoint impossible. Therefore, big data quality dimension focuses more on the dimensions that cover the main characteristics, volume, velocity, and variety. Some other agreed quality dimensions for big data are accuracy, consistency, completeness and timeliness (Taleb *et al*. 2016; Hazen *et al*. 2014).

Big data is very relevant to the energy field. Large amounts of data is generated through measurements of incoming, internal or exiting flows in the energy network (Schuelke-Leech *et al*. 2015, p. 939). The volume of data is increasing with advancements in the key technologies for the energy networks, such as the network communication technologies and the transmission technologies. There is an increasing need to utilize this big data in the operation of energy systems, to increase stability, security, and efficiency. The sections of big data and how they are changing data is illustrated in Figure 13. (Zhou, Fu *et al*. 2016)

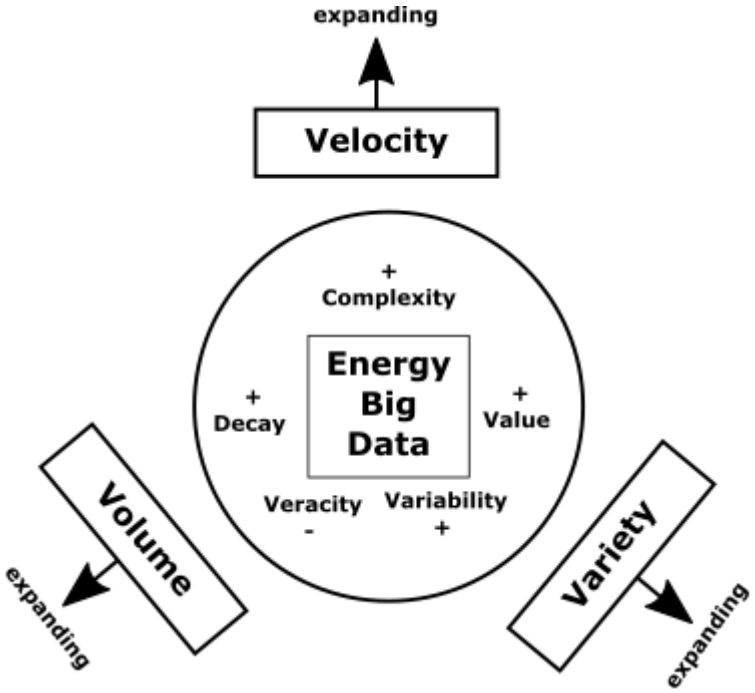

Figure 13. Energy big data explained, adapted from Koseleva and Ropaite (2017) and Lee (2017).

4 QUALITY MONITORING AND CONTROL METHODS

Automatic data processing methods are required in real-time data processing, to retrieve the real-time data from site, according to Vejen *et al*. (2002). Applications that continuously generate data require the ability to make decisions on-line, as the data arrives from the system, especially, because streaming and sensor data is listed as highly volatile data (Ehrlinger *et al*. 2018a). These on-line algorithms work based on the concept, that they only investigate the relevant data and look only once in a fixed order determined by the data arrival pattern. (Garofalakis *et al*. 2002)

Real-time data processing deals with data in real-time or close to it, while being gathered from the system. The requirement of the automatic data processing is to make it possible to utilize said data by the data consumers e.g., further processes in the system like in the energy networks case. Real-time on-line quality control for the data is thus needed. Range and limit checks, step checks for parameter changes, internal consistency checks, missing value and syntax control and comparison between observed and expected values are examples of quality control methods that can be applied in real-time. (Vejen *et al*. 2002)

Data quality criteria are formed based on the assessed data quality dimensions and the requirements set on the collected data quality. They are used to standardize the data quality from multiple, possibly varying, sources (Loshin 2011c). They are utilized to validate the quality of the source and the suitability in the intended use of the data. They can be formed after analysing the requirements of the target process (Loshin 2011d). Detecting low quality data comes from the limit/format violations, and the severity of the these violations, and how many of them are violated, determines the degree to which the quality is lowered. (Chen *et al*. 2017; Kantardzic 2011, pp. 215–218)

A confidence model has been suggested by Ardagna *et al*. (2018) for assessing big data quality. They also conclude that the data quality assessment dimensions are strongly dependent on the type of data and the source, so the algorithm needs to be defined according to target process. Similarly, Sha and Shi (2008) have proposed a consistency model for networked sensor systems, because of the type of data that sensors produce and which data quality dimensions they reflect. The collected data should be accurate

and timely, as they reflect the streaming nature of sensor data. The model observes data quality from three different perspectives: numerical, temporal, and frequency consistencies. These incorporate the possible sources of error in the sensor networks.

Each dimensions can be quantified by one or more quality metrics, that need to be identified to represent the wanted quality requirement (Schneidewind 2005). Some metrics can be assessed by identifying a gold standard reference value, that the assessed value can be compared to, but in real life these are rarely available. Benchmark values for the data quality are thus needed to be gathered from previous high-quality data. (Ehrlinger *et al*. 2018a)

The taxonomy in the data quality monitoring and control is explained in Figure 14. It is formed for this study based on the data quality dimensions presented at Ge *et al*. (2019) for smart grids. The pre-processing phase, where data is received, can deal with the dimensions presented in that phase immediately, as their quality is resolved simply by just a check that determines whether the data is of quality or not. Further quality analysis requires more calculation power/time to determine the quality. Quality flags are given in both phases to explain the data quality.

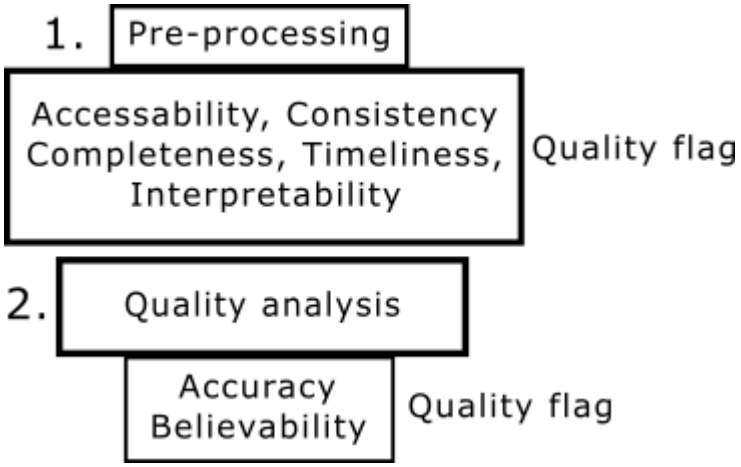

Figure 14. Data quality control taxonomy.

4.1 Metadata

Metadata consists of rules describing the correctness of the data, where the rules are contextual on the type of data. There is no need to deploy data profiling if the data is

accurate. However, this is not the case most of times, which leads to the need for a decision to be made. Is the data or the metadata accurate? If both are incorrect, some fundamental fixes need to be made to establish an accurate foundation to build upon in the first place. Determining data and metadata accuracy, giving the information about deficiencies in data, is done in data profiling (see Figure 15). (Olson 2003, p. 122)

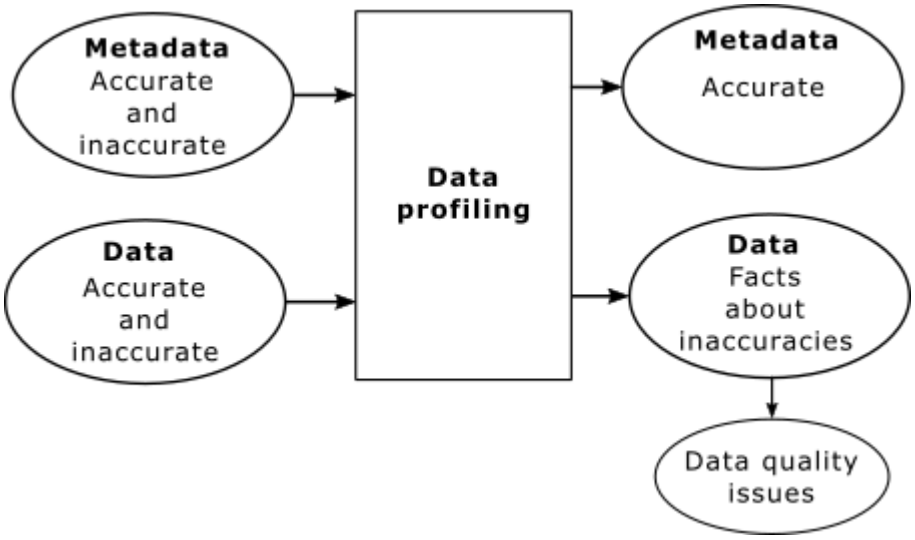

Figure 15. Metadata and data profiling, adapted from Olson (2003).

Depending on the data source, the metadata might contain different things and have a different structure. Generally, the metadata consists of standardized definitions, structures, nomenclature, and determination of existence and other important information about real-world attributes about the data. This can be utilized to assess quality through simple tests to see whether the data quality rules are fulfilled or not. (Loshin 2011c)

Determination of inaccuracies needs a reference in the form of true values. Therefore, metadata is important, as it defines what accuracy is of the data. For example, if the metadata depicts the data comes from a source that is measured outdoor temperature, the data would most definitely be flagged inaccurate if the value would be over 100 degrees Celsius. The initial structure is defined by the available metadata, even though the assumption is that it is inaccurate and/or incomplete. (Olson 2003, p. 124)

4.2 Pre-processing

Data pre-processing in this context are the techniques used to achieve required form, volume and content out of raw data to be input into a data quality assessing algorithm (García *et al*. 2015, p. 2). Here completeness, consistency, timeliness, and accessibility dimensions must be monitored before the accuracy and believability dimensions can be reached. Data volume is usually reduced in real-time applications to improve the efficiency and ease of extracting relevant information from the data (Santhanam and Padmavathi 2014).

What the data consists of is studied in data pre-processing, and that information is utilized to clean the data. Pre-processing is used to give insight about the basic quality aspects in the data, that should be highlighted before further analysis. This information can also help to fix or remove inconsistencies and fill in the missing values using basic statistical methods (Han *et al*. 2012, p. 39). Pre-processing is done on already received data, to clean the data without removing any information from the raw data. Pre-processed data is then sent to the actual data quality assessment. (Zhang *et al*. 2018)

The pre-processing phase of the data quality management is done to prepare the data for further processing, because raw data can be inconsistent, incomplete and include noise. These deficiencies cause problems in the actual data analysis process and in the further processes that utilize said data (Mendel and Korjani 2014). Data pre-processing includes cleansing from duplicate or erroneous data and missing values (Krishnan *et al*. 2016), removing noise and outliers (Siddiqui *et al*. 2020) and tests for completeness, timeliness and accessibility.

Handling copious amounts of data, generated by numerous sensors in the energy network, requires the data-management system to pre-process incoming data quickly. Pre-processing is done to avoid confusion in the operation, by delivering high quality information. There is usually a physical understanding of the energy system, which makes this process easier. (Wolf 2016; Catterson and McArthur 2016)

4.2.1 Completeness

Assessing completeness refers to comparing the number of metrics present in the data to the number of metrics that should be present in the data. (Ehrlinger *et al*. 2018a, 2018b). The completeness data quality score reduces depending on how many data values are missing. They can either be missing values or outliers (Radhakrishnan and Das 2018). Completeness can be further broken down into schema (equation (1)), column (equation (2)) and population completeness (equation (3)) (Michael 2015). Schema completeness is the degree to which required attributes are present, column completeness defines the missing values and properties in the data columns, and population completeness is degree to which real-world reference points are present in the data. The completeness can also be inspected from the interlinked (equation (4)), data instances point of view (Gu *et al*. 2012).

$$Schema\ completeness = \frac{Number\ of\ classes\ and\ properties\ represented}{Total\ number\ of\ classes\ and\ properties}, \quad (1)$$

$$Column\ completeness = \frac{Number\ of\ values\ presented\ for\ a\ specific\ property}{Total\ number\ of\ values\ for\ a\ specific\ property}, \quad (2)$$

$$Population\ completeness = \frac{Number\ of\ real\text{-}world\ objects\ represented}{Total\ number\ of\ real\text{-}world\ objects}, \quad (3)$$

$$Interlinking\ completeness = \frac{Number\ of\ interlinked\ dataset\ instances}{Total\ number\ of\ instances\ in\ dataset}. \quad (4)$$

Missing values need to be imputed if the completeness is not sufficient or errors may ensue as stated earlier. Three different methods for missing data imputation are represented by Santhanam and Padmavathi (2014): imputation by mean, by median and by predicted score (regression). The choice between these suggested methods depends much on the data type and content, as there might be big differences in the variance and correlation of the data points (Acuna and Rodriguez 2004).

Imputation methods have potential error inducing effects. Mean (equation (5)), imputation might distort the distribution of the values, overestimate the sample size, underestimate the variance, and lead to biased correlation. Mean is also affected by outliers, why median (equation (6)) is more natural choice, but imputing a same value

may lead to induced error (Acuna and Rodriguez 2004). The regression imputation will predict the missing value from the prediction curve and thus find out what value is missing or should be present. Autoregressive model (equation (7)), usually used in offline applications, requires training data, while recursive autoregressive model operates with the same principle, only with real-time data (Wang and Makis 2009).

Mean:

$$\hat{x}_{ij} = \sum \frac{x_{ij}}{N_k}, \tag{5}$$

where x_{ij} is the inspected datapoint,
$\hat{x}_{ij}$ is the estimated mean, and
N_k is the total number of datapoints.

Median:

$$\hat{x}_{ij} = \begin{cases} \frac{n_k+1}{2}, \ n \text{ is uneven} \\ \frac{n_k+n_{k+1}}{2}, \ n \text{ is even} \end{cases}, \tag{6}$$

where $\hat{x}_{ij}$ is the mean and
n_k is the datapoint at instance *k*.

Autoregressive model:

$$y(t) = \sum_{k=1}^{p} a(k) * y(t-k) + e(t), \tag{7}$$

where $y(t)$ is the current value,
$y(t-k)$ are the previous values,
$a(k)$ are the coefficient for the AR model,
p is the order of the model, and
$e(t)$ is the noise variable.

Training requirement may have the opposite negative effects in contrast to the mean missing value imputation. As the most likely value is imputed, uncertainty and residual variance are not considered leading to overfitting being a likely problem. Or the training might not keep up with the actual changes, leading into bad estimation. Also, this approach assumes linear relationship between attributes, which is usually not the case in real life.

4.2.2 Timeliness

Timeliness measures the age of the data (equation (8)), and to which extent it is viable to be used in the system. It is largely dependent on the type and target of the assessed data utilization, as different processes require different levels of timeliness. Timeliness is affected by currency, which represents the delay of real-world events being reflected in the data. Currency (equation (9)), relates to timeliness, as it measures the time between the data updates. (Michael 2015)

$$Timeliness = max\left\{0,\ 1 - \frac{currency}{volatility}\right\} \tag{8}$$

$$Currency = t_{storage} - t_{actual}. \tag{9}$$

The freshness of a data also depends on volatility, which is the period for which a certain piece of data remains valid. Volatility is a domain specific value (Ehrlinger *et al*. 2018b). Timeliness is given values between 0 and 1, where 0 means the data is outdated and invalid and 1 means the data is timely. Timeliness, determines whether the data is fresh enough for its purpose. (Zaveri *et al*. 2016)

4.2.3 Consistency

Consistency defines to the extent which extent the data is free of contradictions, which can be present as inconsistent values or attributes in the data. These inconsistencies can cause values or attributes being out of place, out of constraints values, inconsistent correlations or ambiguity in general (Moreira *et al*. 2018). Consistency can also be defined as the extent to which data units and the relationships between them correspond with the system requirements (Xiaojuan *et al*. 2008). Consistency has to be analysed in

respect to the target application, which means that in networks there is a need to assess the specific requirements and features of each application (Sha and Shi 2008).

$$Consistency = \frac{Number\ of\ consistent\ entities\ in\ the\ dataset}{Total\ number\ of\ entities\ in\ dataset} \quad (10)$$

Equation (10) can be further broken down into structural, semantic/internal, and spatial consistency. Structural consistency refers to the presentation of values and attributes being similar across different datapoint, which is tested by comparing the standardized structure to the inspected datapoint. Semantic consistency refers to consistency between definitions for attributes in the inspected datapoints. The meaning and naming of similar attributes should remain consistent. This is checked by going through the attribute definitions in each datapoint. Internal consistency refers to the values in the data remaining within physical constraints and being plausible. This is checked by comparing the value to the physical constraints to see if it falls into the range and by comparing values between similar parameters. (Loshin 2011b; Steinacker *et al*. 2011)

Duplicate values are also a source of inconsistencies and inaccuracies in the process (Olson 2003, p. 183). The goal is to identify duplicates in the incoming data, by comparing their identities. If two or more of the same identities appear in a row in real-time data stream case, the duplicates need to be removed. Even approximately similar datapoints can be potentially removed, if it achieves better efficiency of operation (Batini 2016). There may be slight differences between duplicate data in the metadata, thus making it more difficult to notice, while inconsistencies appear along with redundant and correlated attributes (García *et al*. 2015).

4.2.4 Interpretability

Interpretability refers to the extent to which the content and the properties of the data and metadata can be interpreted correctly (equation (11)). This is dependent on metadata, as it holds information about the data itself (Michael 2015). Whether a machine or an automated system can comprehend the information within data is defined by interpretability. A difference in symbols, notation, vocabularies, terms, objects, data types, presence of undefined classes and properties, and of blank nodes can hamper

interpretability. (Feeney *et al*. 2014; Hogan *et al*. 2012; Pipino *et al*. 2002; Zaveri *et al*. 2016)

$$Interpretability=1-\frac{Number\ of\ uninterpretable\ entries}{Total\ number\ of\ entries}. \quad (11)$$

The method to solve interpretability quality is to solve how much of the data is in an interpretable form compared to all the available data, which can be hampered for example by erroneous notation. (Zaveri *et al*. 2016; Radhakrishnan and Das 2018)

4.2.5 Accessibility

Accessibility or availability (equation (12)) means the extent of which data is obtainable, possible to use, understandable and ready to be used in the system. (See *et al*. 2008; Chen Zhou, *et al*. 2017). For example, if trying to access a website gives the message "website is blocked in your country", it means that data is not accessible/available for use. Data is either accessible, or it is not, which means the data is given a value of 0 for inaccessible or 1 for accessible.

$$Accessibility=\frac{t_{del}\text{-}t_{req}}{t_{dl}\text{-}t_{req}}. \quad (12)$$

The equation (12) describes a time-based situation, where t_{del} is time when the data is delivered, t_{dl} is the deadline before which the data must be delivered and t_{req} is the time the data was requested (Radhakrishnan and Das 2018).

4.3 Analysis

Perfect accuracy cannot be achieved, but some level close to that is achievable. If that achieved level is within the tolerance limits, the system will operate as intended. Beyond the tolerance level for inaccuracy, the system might operate at a level that is acceptable, although more inefficiently, so that the inaccurate data and its source may go unnoticed. After the data inaccuracy rises even further, its usefulness of it drops to zero, as it offers no added value. There is thus a relationship between the usefulness and the accuracy of the data, which can be utilized to determine the "useful" level of data accuracy in an environment under scrutiny. Data in the "good zone" is not 100%

accurate, but accurate enough to give added value in use. This is illustrated in Figure 16. (Olson 2003, pp. 40–41)

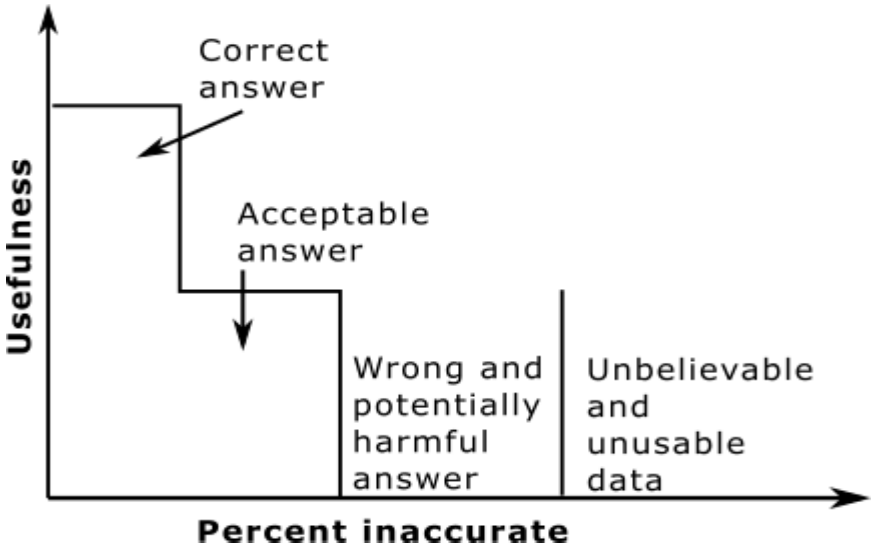

Figure 16. Data usefulness in relation to data accuracy, adapted from Olson (2003, p. 41).

4.3.1 Accuracy

The way accuracy dimensions is solved boils down to the number of data that fall into the defined range of inspection compared to the total number of data. (Radhakrishnan and Das 2018; Even and Shankaranarayanan 2005). The valid range for the data accuracy is expressed in the rules for accuracy (Michael 2015). The rules and the valid range depend on the assessed data source naturally. It can also be expressed as the ratio of the data with and without noise (Zhang *et al*. 2019).

The accurate range of values is defined based on the valid real-world values (Zaveri *et al*. 2016). These values that fall into this range are marked as accurate. The accuracy also depends on the integral value completeness. The accuracy metric can be also addressed by probability tools to investigate the probabilities that the value is correct in relation to the sensor reading. These probability based methods, without prior knowledge of the probability distribution, are able to give objective information about the sensor reading in relation to the predetermined threshold values (Zhang *et al*. 2019).

Accuracy is divided into syntactic and semantic subsections. Syntactic accuracy refers to the valid set of allowed set of values, and semantic accuracy refers to the correct state of the inspected object. For example, inspected set of values has numbers between 1 and 10, while the actual value is 7. In this case 11 would be incorrect in syntactic accuracy sense, while semantic accuracy would be incorrect if the data shows anything else than a 7. Semantic accuracy is difficult to assess, but it can be done through dependency rules and higher weights on critical data values (Fürber and Hepp 2011).

Syntactic validity is also presented as a part of accuracy by Zaveri *et al*. (2016), while citing the given definition for it by Flemming *et al*. (2011). Syntactic accuracy measures how the assessed data is correct in respect to the definition model (Michael 2015). The syntactic validity consists of correct syntax and vocabularies. The syntactic accuracy metrics are inspected through validators, syntactic rules, and datatype inspection.

$$Accuracy = \frac{Number\ of\ valid\ data}{Total\ number\ of\ data}, \tag{13}$$

$$Accessibility = \frac{t_{del}\text{-}t_{req}}{t_{dl}\text{-}t_{req}}. \tag{14}$$

Syntactic accuracy is calculated with explicit range checks and assigned rules in relation to the specified valid range (Wienand and Paulheim 2014). For the syntactic accuracy to be of high quality, the datapoint needs to fall into the pre-set valid ranges, or the value needs to follow valid value specific rules (Fürber and Hepp 2011). Also, the data must be of correct syntax to pass as high quality data (Zaveri *et al*. 2016). Equation (15) shows threshold accuracy,

$$Threshold\ Accuracy = \begin{cases} 1, & T_L \leq X \leq T_H \\ 0, & X \geq T_H \text{ or } X \leq T_L' \end{cases} \tag{15}$$

where T_L is the lower threshold limit for the valid range,
X is the inspected datapoint value and
T_H is the higher threshold limit for the valid range,

while equation (16) shows the concept for rule accuracy,

$$Rule\ Accuracy = \begin{cases} 1, X \rightarrow R \\ 0, X \neq R \end{cases}, \tag{16}$$

where X is the imputed datapoint value and

R is the accuracy value rule to be satisfied.

Semantic accuracy is calculated by outliers in relation to accurate data, which can be done by distance-, deviation-, and distribution-based methods (See *et al*. 2008; Bizer and Cyganiak 2009) or using statistical distributions to assess the correctness/accuracy (Paulheim and Bizer 2014). Semantic accuracy can also be assessed within the valid range by the dependencies between the values within the different properties of the data (metadata) (Fürber and Hepp 2011) and by comparing two or more values of the same source (Kontokostas *et al*. 2014). Principal component analysis (PCA) validation is used for semantic accuracy because it is a well-established validation method, and the energy network contains sensors with redundant data.

According to Smith *et al*. (2012), uncertainty can be quantified using probabilistic methods on sequentially correlated readings. The uncertainty can be solved using a statistical model, fuzzy set model or random-fuzzy model (Timms *et al*. 2011), where the final error is defined by the manufacturer given accuracy value of the sensors. Calibration and fouling are also variables, that affect the sensor reading uncertainty, and they can be included within the uncertainty determination, but the information for the calibration and fouling might be hard to come by in a real-time system.

Klein and Lehner (2009) have suggested separating the data into synopses and investigating the uncertainty and quality over different dimensions. The uncertainty across the data values is assessed for individual datapoints by accuracy and confidence values for measurements and a threshold function. This defined range for uncertainty for the datapoints can lead to false positives or false negatives when addressing whether to get rid of or keep the datapoint. The confidence value for the data is defined by a formula, that utilizes the past confidence value plus the new statistical error.

The validation of sensor signals is done to avoid process disturbances by undetected errors in the sensor function and to locate and fix them. The sensor fault ranges from deviation from normal to complete failure, and the validation procedure must recognize

these in the observed signal patterns. Fault analysis is done through statistical and mathematical means through regression from predicted or measured variables. (Rosinés 2007)

One potential method for assessing accuracy of redundant sensor signals is Principal Component Analysis (PCA). It utilizes the correlated variables from multiple sources and reduces them to uncorrelated principal components. Principal components are eigenvectors, in which the eigenvalues are set in a descending order (Sen *et al*. 2019). The eigenvalues describe the variance between the variables, so that the best variance describing principal components are thus chosen to approximate multivariable datasets in reduced dimensions, while preserving most of the characteristics of the original data (Ballabio 2015).

Hotelling's T^2 statistic is a measurement of variation, being sum of normalized squared scores and Q statistic is a measurement of residuals and how well the samples conform to the PCA. Hotelling's T^2 statistic and Q statistic can be utilized to identify anomalies in the dataset, which are outside the confidence bounds of these statistics. (Ballabio 2015; Bro and Smilde 2014). This function of the PCA analysis can be used to spot erroneous values in sensors with redundancy efficiently (Rosinés 2007).

The data from multiple similar sensors is correlated with each other, which helps in detecting fault by redundancy. Residuals can be utilized to detect the faults in the measured data distributions, as residuals will be non-zero in the presence of irregular disturbances, at least in theory. Noise will be present in real-life operation, so methods that consider the deviations from the theoretical values and unmodelled dynamics are used. Such methods as adaptive error threshold and statistical tests. (Rosinés 2007)

Squared prediction error (SPE) is utilized to test and detect for faulty signals from multiple similar sensors. In sensor validation index (SVI), SPE values are used to find the faulty sensor. The SPE values are calculated with the equations (17)–(20),

$$x_t = y_t + e, \tag{17}$$

$$C = P*[(P'*P)-1]*P' = P*P', \tag{18}$$

$$e = x_t\text{-}y_t = x_t\text{-}\left(P*P'*x_t\right) = \left(I\text{-}P*P'\right)*x_t, \tag{19}$$

$$SPE = \|e\|^2 = \left\|\left(I\text{-}P*P'\right)\right\|^2 \tag{20}$$

where x_t is the new data sample,

y_t is the projection variable containing all the variations,

e is the variable containing residuals from the projections,

C is the projection matrix,

P is the loadings matrix,

I is a unit vector and

SPE is the squared prediction error.

The residuals are compared between each of the sensor measurements and the faulty reading, until the residuals drop close to zero. This indicates that the faulty sensor has been spotted. (Rosinés 2007)

4.3.2 Believability

The concept of believability is dependent on accuracy, as accuracy is the dimension that describes similarity with reality. Believability describes the extent accuracy value is seen being true or rational. Other dimensions also describe accuracy in their own way too, mostly the structure and content, but believability questions the source of the data, which is not done by any other dimension. Methods questioning the integrity of the data source focus on confidence models and fuzzy methods (Shekarpour and Katebi 2010), that utilize statistical probabilities to determine how trustworthy or believable the incoming data is to begin with. What is suggested by Moossavizadeh *et al*. (2012), is to utilize the normalized data quality values from each of the assessed dimensions and get a value for believability that way. (Pradhan 2005)

Believability or trustworthiness is an important dimension when dealing with big quantity. It is the extent the data is believed to be true, credible, real and correct (Pipino *et al*. 2002; Zaveri *et al*. 2016). The believability data quality is calculated based on the provenance of information. It gets a value between [−1,1], where 1 is total belief and −1 is a total disbelief (Hartig 2008). The suggested method for calculating believability by

Shekarpour and Katebi (2010) consists of two algorithms. The first one for propagation utilizing statistical techniques and the second one utilizing max-weighting mechanism.

Statistical fuzzy max-weight method, presented by Shekarpour and Katebi (2010), could be utilized to calculate the believability value based on the probability that the inspected datapoint is believable based on the previous datapoint values. The believability can be presented in approximate reasoning through fuzzy models, combining the accuracy values into the believability assessment by calculating the probability that the accuracy value is wrong. Precision and recall are metrics defined to assess the trust between two nodes in three different states of trust; trust state, distrust state and general state, which is the combination of the previous two.

Different inspected data quality attributes/dimensions have different weights, so they need to be balanced (Han *et al*. 2012). For the data to be evaluated similarly across all dimensions, the data needs to be normalized to fit an specific range, for example [0,1] (Baskar *et al*. 2013). Normalization is listed as an inspected quality dimension by Ehrlinger *et al*. (2018a; 2018b), as normal forms to avoid inconsistencies, redundancies as well as get rid of anomalies.

4.4 Quality flagging

Data profiling is a tool for assessing data quality and determining the future use of said data. Data flagging is part of data profiling. It allows to categorize data based on quality characteristics, which makes using high quality data and filtering out bad quality data easier and more effective. Metadata can give information about characteristics within data, that would otherwise be omitted. Analysis of the data based on pre-set quality conditions, is done in the quality monitoring phase that determines the quality flag, for example rules that the data needs to fulfil. The quality monitoring test result shows how the data differs from the set quality standards. (Olson 2003, p. 20)

As illustrated in Figure 14, data flagging is done in two parts. Firstly, the pre-processing phases include simple quality checks that are either limit or format tests to determine the data flag value. After this, the accuracy is determined, which is affected by the data flag values of the pre-processing quality dimensions and the past accuracy values. If the last

accuracy quality flag was good, simple difference between the current and previous value is checked to discover sudden changes, but if the last flag was bad, the accuracy is checked with moving median of set inspection window size.

Data flagging is a improve the quality and the reliability of the data coming into from the energy network by quantifying the incoming information quality. The data flagging is done to raw data coming into the system, so that best possible decisions can be made with the most relevant data available. Quality flagging thus affects only the available data.

Data flagging is a way of using automatic screening algorithms to provide swift and approximate information about the data quality and attributes (Geuder *et al*. 2015). Data flagging does not alter the data. It gives each observation, or data point, an attribute about its trustworthiness in relation to quality assurance (QA), (Shafer *et al*. 2000). The data is assigned with flag values of different parameters (data quality dimensions) to give deeper understanding of the data quality and the possible shortcomings of the data.

A variety of control flags are needed in the quality control process, as many different methods are in use and the operation is in real-time. Each data element needs to be flagged to aid in the systems decision-making. The flag information should contain quality level, quality method, type of error and possible correction. (Vejen *et al*. 2002)

The data flag structure (Table 1, p. 56), shows each data quality dimension and the data flag values they can give. The data flag value of 1 means that the quality of the inspected data is bad, it is inaccurate, and it should not be used. The data quality flag of 3, means that the quality is lacking, or that some element of the data has been estimated. Data quality flag of 5 means the data is accurate and it passed the quality test.

4.5 Synthesis of methods

The chosen and observed data quality dimensions are present in the chosen methods to observe and test the data quality. In this way, the explanation of the meaning of the data quality in general and the meaning of individual data quality dimensions come together in a meaningful sense, that is relevant to energy networks. The approach must take into

consideration the high volume of data, thus focus more on certain aspects of the data quality, which are well found in the chosen data quality dimensions.

The methods need to be simple, yet effective, because of the real-time dynamic of the data stream. Therefore, the chosen methods are mostly simple tests and validation through PCA, which is a mature analysis method. The order of methods is following: First the tests for timeliness, completeness, consistency, accessibility, and interpretability are made, with possible imputations for missing values in the completeness check and removal of duplicates in the consistency check. After that is done, the syntactic accuracy can be measured followed by semantic accuracy validation. It is paramount that the pre-processing quality checks do not remove any information from the data, or otherwise the quality of the data cannot be monitored accurately.

Within each quality step and test, a quality flag is assigned for the datapoint to give information about the intrinsic characteristics of the datapoint. These quality flag values can either be binary or a range of values, depending on the subject under quality monitoring. As an example, data is either accessible or not, thus it gets a binary quality flag value, but accuracy can range from 0–100% giving it more possible quality flag values. The quality flagged data will have relevant quality information about itself, and thus can be utilized with better weighing in the system. Bad data can be avoided better, and good data usage can be focused upon.

5 SIMULATED CASE: QUALITY MONITORING OF TEMPERATURE

To solve the research problems of this thesis, the data quality management/control topic must be tied to the energy field and further to energy networks. To achieve this, a simulated algorithm for this case was generated together with simulated data to test the algorithm in real-time operation. The simulated case consists of measured temperature signals, that are quality monitored.

Generating data, spreading it into different observable sections, imputing errors into the data, fixing/imputing errors and missing values and finally flagging the data quality, is the framework of the algorithm. Choosing correct methods to simulate errors is a crucial part of the whole simulation. The general structure of the algorithm is presented in Figure 17. The phases seen in Figure are explained in the further sections. The algorithm is divided into subprocesses, where the most prominent ones are data generation, error introduction and quality flagging. These actions are in sequential order:

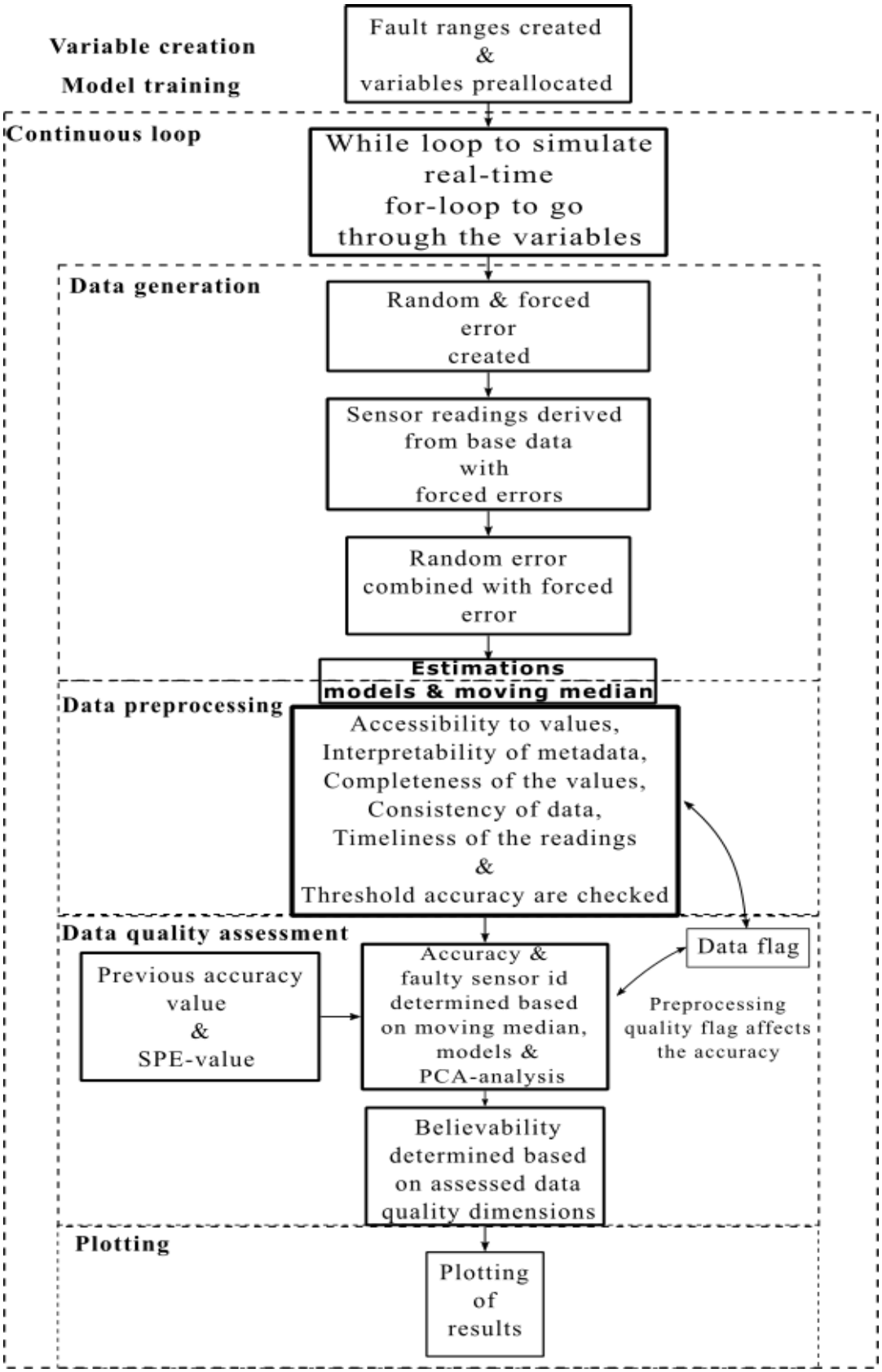

Figure 17. General structure of the algorithm.

5.1 Data generation

To begin with, data is generated to process and to observe the function of the algorithm. This data is made to mirror real-life scenarios, namely heat production and its changes throughout the day due to outside temperatures. The basic range is set according to some basic temperature measurements observable in Finland, and randomization is induced to this range to simulate the changing weather outside.

Outgoing heat from a district heating plant correlates with the heating power and electricity price, so the transfer between observed units is not difficult. Heat production supply and demand both depend on outside weather, which usually follows a trend. The day is warmer, because the sun is shining and less heat is required to be produced, while the night is colder as there is no sun. The winter of Finland, modelled/simulated here, is especially like that, with little sunlight hours in a normal winter day.

Data for this data quality monitoring is done according to the described district heat-demand trend. The changes in the required heat supply, between different times of a day, can be simulated with added random variation by imitating the average district heating supply temperature-graphs, found in Timonen (2018). The continuous trend of the heat demand means that there is a smooth transition between days, as the data should include wide variety of scenarios to test the function of the data quality algorithm. Depending on the scenario, flexibility is required from the algorithm to be able to be utilized on data quality control in different parts of the energy network.

The base trend has randomization through noise, but not in the shape of the trend, to simplify the analysis. The trend is meant to repeat, having different randomization in each loop iteration, which further simulates the unexpected noise from sensors. The noise is included also in the data flagging process, as the goal is to process the data as raw as possible to then give the choice to keep or drop the data in the further phases of data quality management.

Approaching this subject from a real-life standpoint, the algorithm is based on a loop, that runs one point at a time. This allows to simulate real-life data streams, which are tied to real-time and produce one datapoint at a time. It also makes the algorithm to be

flexible, by changing the step time of the produced datapoints, meaning the time between two generated datapoints. In addition to the value of the datapoint, the time, date and name of the sensor are given with the sensor reading. The time of the simulation is one week, divided into 3-minute increments.

Forced and random errors are utilized to represent the errors, which may appear in everyday operation, due to system dynamics, and random errors are utilized to simulate the unexplainable errors in the sensor readings and in the system in general. Combining these two together, makes up a simulation that requires a robust ability to respond to errors. Including various errors that could be countered from an erroneous sensor reading. The generated raw data is displayed in Figure 18.

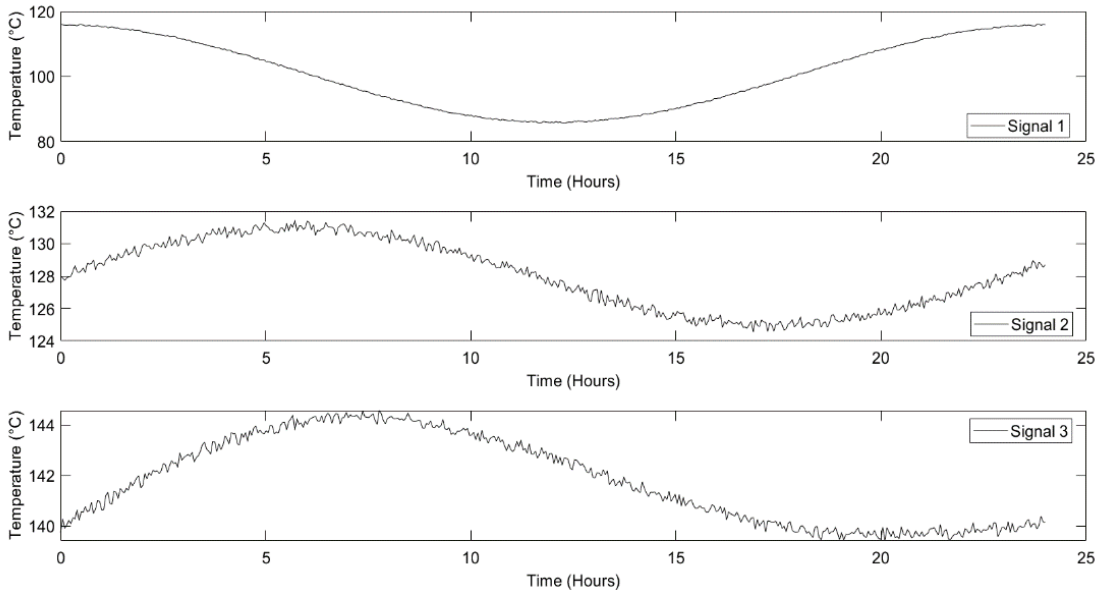

Figure 18. Generated raw data, signals 1, 2 and 3.

Data flags need to be given to the incoming datapoints as they are received, or else they will be too late to help adjust the decision-making. Normally the data quality monitoring would be a separate entity to the process producing the data, but to demonstrate the data quality monitoring in real-time and with simulated errors, it is all done in the same algorithm. In a real-life scenario, this kind of approach would require a high level of optimization, adaptability, and historical knowledge from the data, which is achievable with enough resources at one's disposal.

5.2 Introduction of simulated errors

The variables utilized within the scale of the algorithm and the base for the data are produced before the actual loop to present the starting point of the whole simulation of data quality monitoring. Doing this prevents the algorithm from generating the same variables again and again when the loop resets, and the forced and random error can be generated in the very beginning of the loop. To represent the forced errors, placeholder datapoints for the forced errors are produced, which allow for the errors to be seen in a wanted range of the data loop.

The loop repeats itself a predetermined number of times, to represent the daily variation in the measured data with some minor induced noise and random variation to represent the noise and variation present in real-life applications, described in Figure 20. The loop repeats itself again, during which the data quality determination process starts again from the beginning. To avoid issues with the loop starting again, the forced error and random error range do not include the very beginning or the very end of the point range, but this could be done by utilizing variables that store the values from previous iterations.

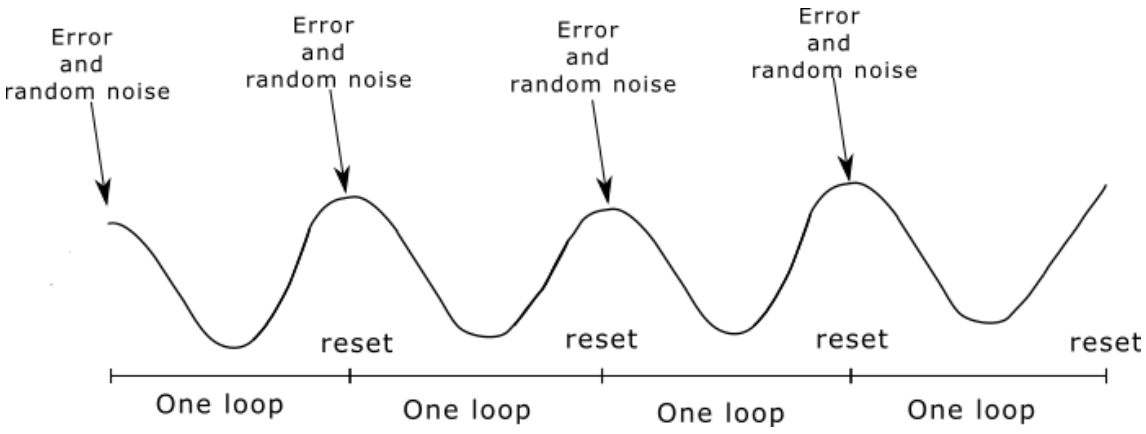

Figure 19. Loop and induced error explained.

Once the data is generated, one point at a time, it is divided into different sections representing number of sensors measuring temperature for signal 1 and only one sensor for signals 2 and 3. Total of eight sensors are utilized, each with different forced errors,

and redundancy between each other. This approach provides possibility to work with identifying, imputing, and flagging low quality data, to better utilize the high-quality data. The goal is to find methods to identify potential issues in data quality and ways to deal with them.

Simulating real-life, the errors are introduced into the divided sensor readings, into known locations, but also at randomly, to test performance the algorithm. The known errors are introduced to test whether the algorithm can identify variation from the mean and the generated base-data and if not, what can be done to make the identification possible. Random error amount is 5% of all the inspection range, and the imputation of random error prevents imputation of two consecutive errors, to simplify the simulation. Random errors are introduced to test out the data quality monitoring part of the algorithm. Erroneous data is likely present also in real-life data, so the algorithm should be able to handle it either by replacing missing values or flagging these data points to restrict their further usage.

Random errors occur in real-life applications unexpectedly. The data quality monitoring system should however be able to respond to these kinds of errors in a timely and efficient manner to keep the system running, by giving accurate data flags. Seeing how the randomized errors affect and distort the algorithm makes it crucial to build the algorithm in a robust and resolute manner, updating it as new problems arise.

The value of individual datapoints is determined based on the actual base data in addition to the forced error. The value in a certain datapoint is one of the most important aspects of the observed data quality, as it relates to the real-life system state. It needs to be a trustworthy and accurate representation of the real-life value, or otherwise wrong decisions are inevitable.

The errors introduced to the known fault-ranges (datapoints, where the error is activated) only affect the value of the data points, but the random errors effect not only the value, but the metadata as well. The metadata holds descriptive information about the data, which is useful in data monitoring. If it is erroneous, it lowers affects the trustworthiness of the data value, even if it might be accurate. This kind of randomness in the errors require real-time or in-the-same-step kind of reaction as lacking or

inaccessible data cannot be used and might lead to wrong decision in lack of proper, trustworthy, and high-quality data.

Firstly, many of the used variables need to be preallocated, so that the algorithm would run smoother/faster. It sets up the framework for the algorithm to operate, as many of the variables change size with each iteration of the loop. It is here, where the function of the algorithm can be changed by changing values on variables, length, logical value placement etc.

The length of one loop is specified with ‘time’ variable, that is 24 hours divided into 480 increments (3 minutes). It determines the size of most of the other variables and the placement of the forced errors. The forced errors are a collection of logical positive and negative values placed on the range of the ‘time’ variable, with different multipliers. There are 10 subsets in total, one of them making up 10% of the size. The forced error is fed into the range including the 2nd and 8th subset and every subset between them, except for sensor 5, that has complete failure error. These fault ranges turn on and off the forced error in known locations. This is done to make it easier to locate and interpret the errors, and to see how the algorithm responds to them, Figure 20.

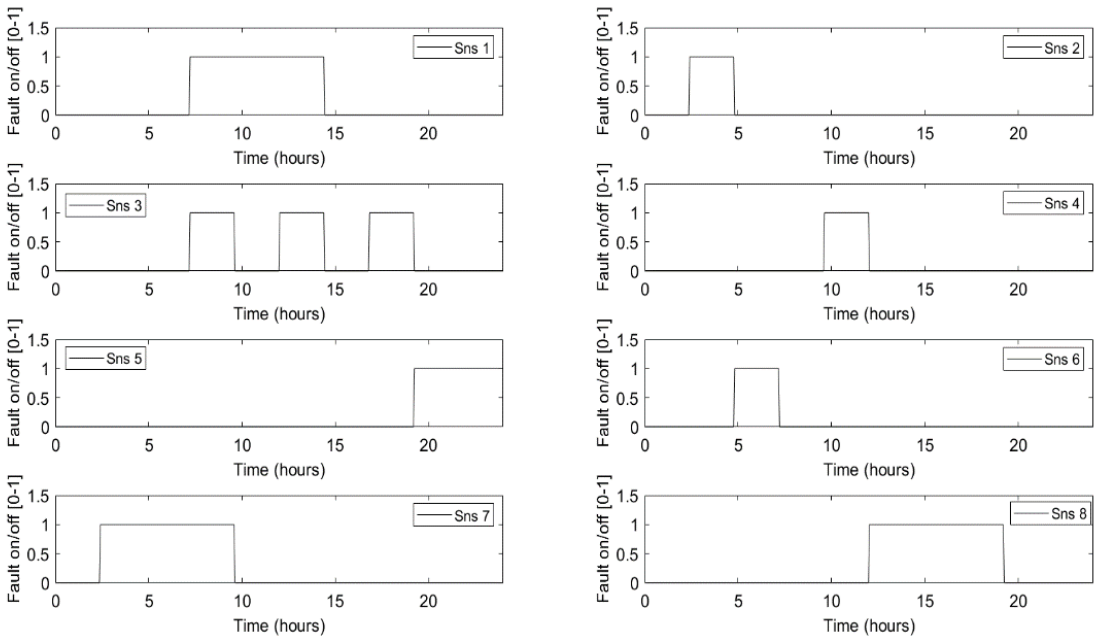

Figure 20. Forced fault ranges of each sensor, logical on/off.

Random errors are given based on a randomized condition-based system. The error can either be empty or NaN (Not a Number), representing missing and unreadable/uninterpretable datapoint, respectively. There is a set number of errors, that are introduced into the inspected variables, around 5% of one full loop rotation datapoints. The random error introduced is either empty or NaN, but not both. The condition based random error imputation makes sure that either one or the other is true, by checking that the condition for imputing an error is true, and that the other condition is not already fulfilled at that datapoint. Both the empty and NaN errors generate a randomized range of errors in the beginning of the loop, and thus there might be some overlap between them, which must be avoided.

The range for these random errors is defined to be from 0 to 80% of the loop range of the datapoints. Random errors distort the algorithm, if they are present at the beginning or the end of the cycle, which is a problem that needs to be delved deeper into. This is due to seeing how the data quality algorithm will react to the forced permanent errors and to ease the inspection.

The generated data is separated into 8 different sensor variables, 6 in the redundant signal and 2 individual signals, each with their own forced faults, fault ranges and names. The idea is that each sensor deviates from the ‘correct’ value in a different way, so the effect of low quality induced by different kind of errors and how they are noticed by the algorithm can be monitored. Errors include drifting, dead value/malfunction, random malfunction, and bias (random deviation from the mean). The locations of the errors are visible in Figure 20.

Sensor 1 error is a bias error of 10, sensor 2 error is variation taken out of the whole signal 1 showing up at randomized time increments at the fault range, sensor 3 error is random standard deviation error, sensor 4 error is bias error with random standard deviation, sensor 5 error is malfunction error and sensor 6 error is drift error. The data itself has noise inputted into it to represent the measurement noise present in industrial applications. In some sensors, that noise is strengthened to a larger degree in their respective fault ranges. Signal 1 includes all the errors (Figure 21.).

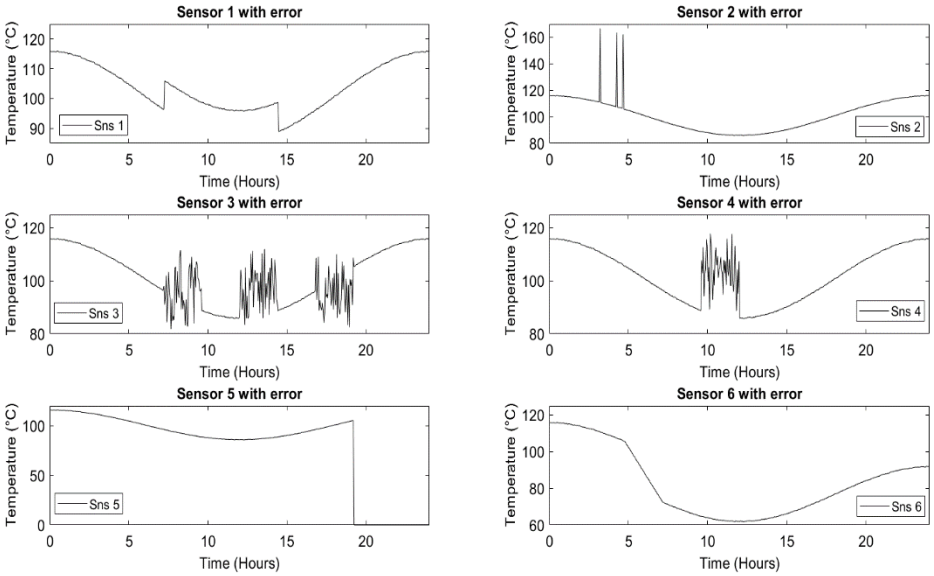

Figure 21. Signal 1 redundant values with forced error.

Simulated signal 2 includes drift, like seen in Figure 22 below. The drift is realized in the beginning of one loop, like seen in Figure 21, which leads the values of the sensor being off for the reminder of the round. The location of the drift is seen in Figure 20. The noise is ±2% of the sensor reading. The drift is a cumulative sum between 02:24:18 to 09:34:12, with a gain of 0.05 every 3 minutes.

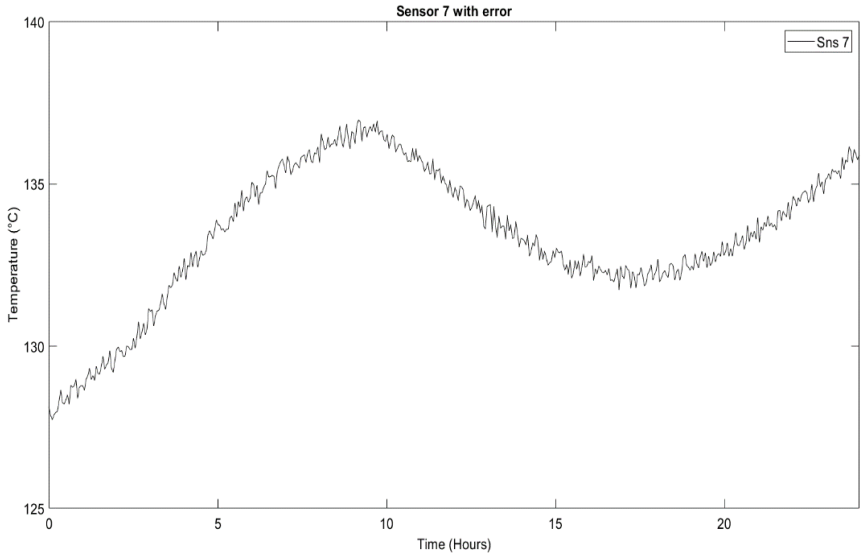

Figure 22. Values of Signal 2 with forced drift error from 02:24:18 to 09:34:12.

The signal 3 includes bias error, Figure 23. The range is clearly seen, as the shift is noticeable. The noise level here is set to ±1% of the value. The location of the fault is seen in Figure 21. The bias is 5 degrees Celsius.

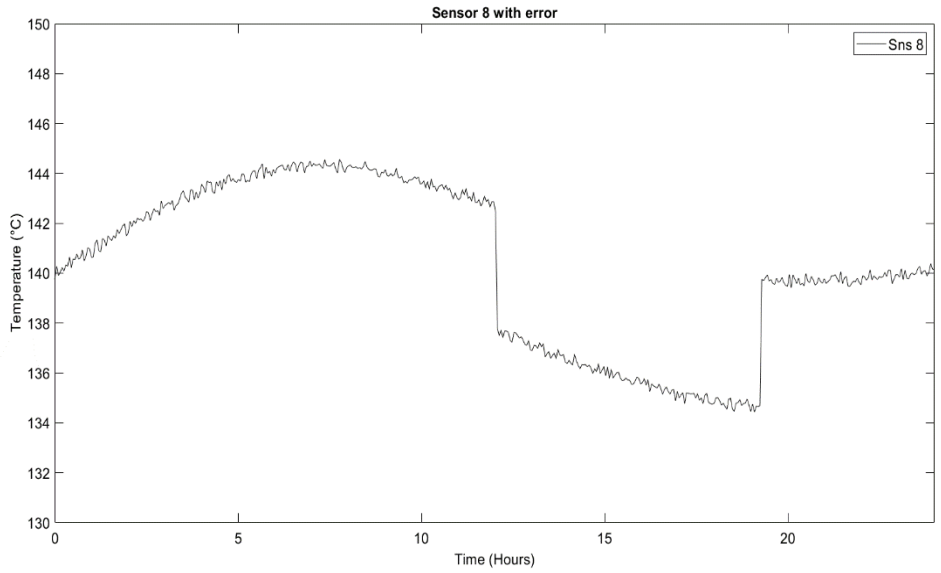

Figure 23. Signal 3 values with forced bias error from 12:01:30 to 19:11:24.

5.3 Estimation of missing and erroneous data

Solving the errors induced into to the inspected data is an additional aim of this study. The target is here then to make the selected approach as simple as possible, and to make it able to function in real-time, without the need of an offline application. Contradicting this idea, a model is trained to predict the next step in the imputation step. To achieve this, an autoregressive model (equation (7)) is utilized together with a recursive autoregressive model. The autoregressive model requires training data, while the recursive model utilizes the incoming data to make predictions of the value. Both models are susceptible to error due to the erroneous data fed into them. The chosen models are similar, the only difference being the continuous training of the recursive model.

One-step-ahead prediction and recursive prediction estimate the current real values in the sensors, should the data quality be low, thus unusable. Recursive autoregressive

model is used to estimate the current value and autoregressive model is used for the one-step-ahead prediction. The autoregressive model requires training data, which is generated with an assumption, that there is understanding of the system behaviour through historical data. If all the received values are flagged for bad quality, the model is used as a backup to feed an estimation forward.

The one step-ahead autoregressive model needs past data to be imputed into it, so the erroneous data present at the time of the model training needs to be filtered. This filtering is done through limiting the fed data by a certain percentile range. This percentile range goes from 40 to 60, which means that the extremes are not considered, and more accurate training data is achieved. This is especially true for the first signal, that assumes redundancy between the different sensor readings. The sensor readings with extreme errors should be filtered out from the mean, so the errors do not shift the estimation.

In addition to the models, moving median values are also calculated for each of the signals. It gives an approximation of the value based on the previous values, that in theory should not differ much from the actual value. The redundant readings are useful in utilizing moving median, as a median value can be gathered from everyone moving median values, (equation (21)). This is more robust in the value accuracy, even if the incoming data is erroneous, if some of the reading values are accurate.

$$Moving\ median\ estimate = \frac{\{movMed(n+1)\}}{2}, \tag{21}$$

where $movMed = \text{median}\{QC(t\text{-}d,j),\ldots,QC(t,j)\}$,
n is the number of elements,
$QC(t)$ is the current sensor reading,
d is the moving median window (here 5) and
j is the sensor reading index (for signal 1, there are 6 sensor readings).

The increased noise is used deliberately to try to account for the randomness induced within the algorithm errors. The training data is generated only once in the loop, while

the noise changes at every loop iteration. The training results for each signal can be seen in Figure 24. The fit (MATLAB® *compare*-function) for signal 1 is 73.69% accurate, which is a moderate to low fit, as the shape of the signal is not complex. Increasing the training does not seem necessary considering the results. The training fit for signal 2 is 60.4% and 64.93% for signal 3.

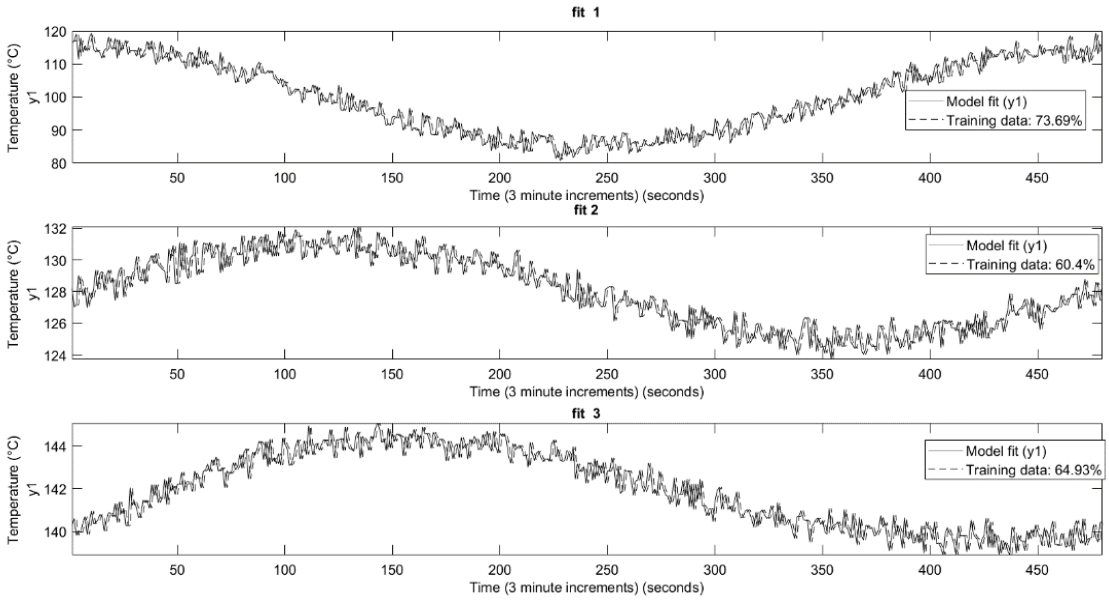

Figure 24. Training data for signals 1, 2 and 3, and AR-model fit results for signals 1, 2 and 3.

5.4 Monitoring of quality dimensions

The data from all the sensors in gathered into one inspected variable after the forced error is introduced and the random error is imputed. This happens one iteration at a time mirroring the online data quality monitoring aspect of this algorithm. This variable then proceeds to the pre-processing phase of the algorithm where the data quality aspects that can be determined by a simple check are done.

The results of the pre-processing data quality checks and their flags (Table 1) can spot almost all the errors while the data comes in, as the checks are quite simple. If the checks did not give a bad quality flag when encountering a bad quality data element and impute an estimation of the missing or uninterpretable element, the rest of the quality

assessment would not be able to function. The pre-processing checks sort out the data in a manner of usability before checking for the accuracy validity. This is in a way precursor for the data dimension reduction, which is generally done after the data quality assessment (flagging). The work is done in a way for the other methods to know what data to drop and which data to include. Thus, it is important, that the pre-processing data quality checks function at almost 100% rate of spotting errors and handling them in a way, that the system utilizing the data requires.

Table 1. Data flag value structure.

Quality dimensions	Given values
Accuracy	[1,3,5]
Believability	[1,3,5]
Completeness	[3,5]
Consistency	[1,5]
Interpretability	[1,5]
Timeliness	[1,5]
Accessibility	[1,5]
Threshold accuracy	[1,5]

The data quality control/monitoring system must have a clear structure, so it can be built and used. The data quality dimensions' monitoring (Figure 25) begins with receiving the raw data, after which the data quality tests are performed upon it. These tests give an idea about the general structure and content of the data, upon which some actions may be necessary to be implemented, if the content or structure is not satisfactory in the sense of quality.

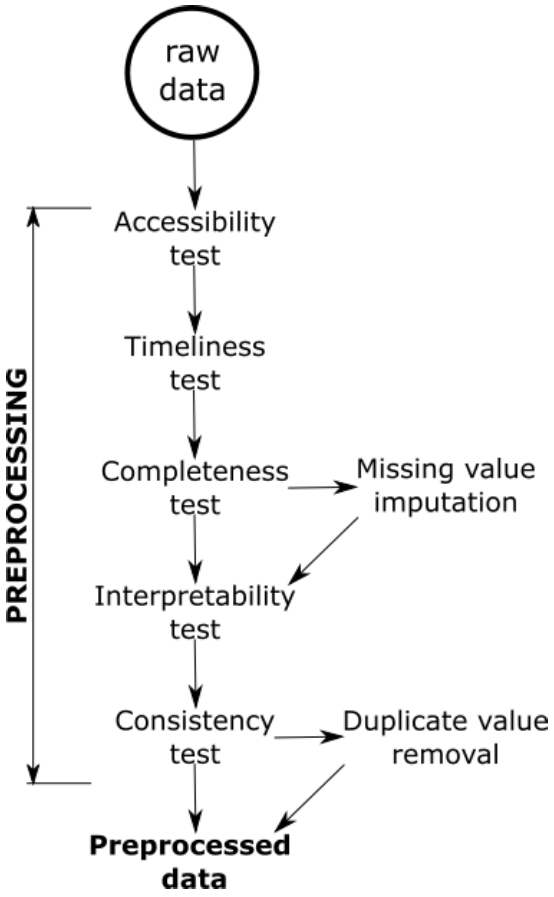

Figure 25. Data quality monitoring process – Pre-processing phase

Once these quick tests and possible small fixes have been implemented, the data can be thought to be in a wanted form to make further analysis on. This further analysis requires, that all the mandatory elements are present, thus making it logically the latter step in the data quality monitoring (Figure 26). Validation of the datapoint gives information on whether there may be an erroneous value, by comparing multiple signal values from similar sensors. Validation of the datapoint values gives information on whether there may be an erroneous value, by comparing multiple value readings from similar sensors.

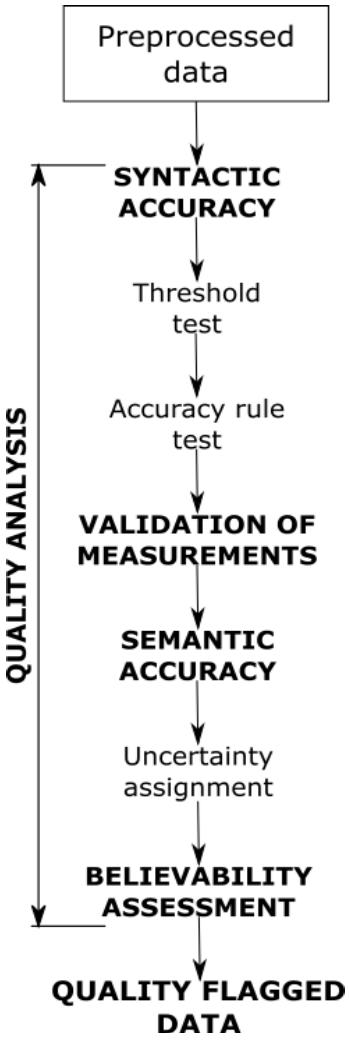

Figure 26. Data quality monitoring process – Analysis phase

Accessibility

Accessibility is the notion, that the data is accessible in the first place, thus of quality for use. To check this data quality dimension, the data value elements are checked for NaN-values for each sensor, which gives out either good or bad data quality flag: good meaning, that the data is accessible and bad meaning that the data is not accessible. Inaccessibility is simulated through the randomized data errors by imputing NaN-values into the data elements, which shows if the data can be accessed for further use or not. If NaN-value is encountered, it is replaced with the better imputed value between moving median and the recursive autoregression-model. If imputations need to be made at this pre-processing step, the data element is judged less believable.

Interpretability

Interpretability deals with metadata. Whether the metadata of a single data-element can be interpreted by the algorithm or not is important to get an over-view of the data quality. Data flag of interpretability does not hinder the value reading in said data element, but it rather gives more accuracy in determining the data quality in the future pre-processing steps. If the data quality flag is bad in interpretability, it needs to be fixed, which is done by replacing the value based on historical data. The uninterpretability is presented as NaN-value from the random error imputation, as in accessibility, and only concerns the two metadata elements in one sensor reading, the time of the capture and the name of the sensor. If NaN is encountered, the time will be replaced by the previous readings time plus the measurement interval. The name is replaced by the name in the previous reading from the same source. When these imputations are done, the data quality is known to be lowered and thus the believability is affected.

Completeness

Checking if the inspected data includes all that it should, is dealt with in the completeness pre-processing step. The datapoint cannot be believable if it is not complete and thus needs actions to be taken upon it. As with the previous checks, the ‘empty’ values are imputed by the random error imputation. The empty variable error is simulated by introducing ‘e’ into the gathered data elements at random, which allows for the algorithm to recognise that the data element is empty. Problems arise with the algorithms function, if such step is not used and the data element is left empty, which beckons further inspection into the algorithms function.

As the ‘empty’ data elements are encountered, one of three actions is taken, depending on in which section of the data elements the ‘empty’ error is detected. If the sensor reading is missing elements, the missing element is replaced by the best imputation value available. If the time of the reading is missing, the imputation is the previous time plus the sensor reading interval. If the name of the sensor is missing, it is replaced with the same name as the previous reading from the same source. These imputations are crucial for the algorithm to gain an idea of an accurate state in the system, whereas data

flagging gives insight and guidance to question the believability of the data. Missing values make the data unusable, thus necessitating that there is at least something to be used in the further checks that utilize, moving averages/moving medians for example, so as to not give erroneous values too much weight.

Consistency

Consistency deals with the integrity of the continuous data element stream. The idea in this case is to check if the data elements are consistent from one to another and this is done by inspecting the difference between two or more subsequent sensor readings and the data elements within them. If there is a change in the sensor name, recording time or the reading value, that is deemed inconsistent, it will be quality flagged for bad data quality. The concept of consistency is subjective and differs with the inspected system, so the thresholds for sensor value and time of the readings differ from system to system and the correct thresholds need to be determined. The threshold for consistency could be dynamic, depending on the inspected system state, but in this study the threshold is static, in order of simplicity. As mentioned above, the sensor reading is checked for differences of ±5% to the previous reading. If this threshold is exceeded, the data value is determined inconsistent.

The consistency check was to assess, whether the data element included the consistent form and style determined for the data. The metadata checks were done to see if the name was in a correct form and length and had no empty spaces in-between, while the time consistency check was to determine that the time/date format was correct. These checks are done according to the content of the generated data.

Timeliness

The sensor reading time between increments is monitored in the timeliness pre-processing step. The time interval between recorded sensor readings is usually set as a constant, for example every three minutes, and deviation from this constant time difference means bad data quality, because the reading is not to be trusted. The lowered data quality might be due to the reading being of old reading, thus it is not accurate anymore. This automatically means, that the inspected datapoint is not believable, as the

time of the reading is not within accurate limits. In the algorithm, the timeliness check is performed by comparing the difference between the current and the previous reading. If that difference is within the set limits for timeliness, the data quality flag is good, but if the difference is greater than the set threshold, a bad data quality flag is given. The algorithm simulates the time in seconds, but that can be translated into any amount of time per increment, by adjusting the number of elements in one loop. For example, if the interval is 3 minutes, the loop length would be adjusted to $\frac{60*60*24}{180} = 480$, which is repeated seven times to represent a week, giving 7*480 = 3360 datapoints in total.

Accuracy

Accuracy is split into two inspected dimensions, where one is a standalone check, and the other is a test, that is affected by other data quality flag results given earlier in the loop increment or in previous loop increment. Threshold accuracy test whether the value goes over the predetermined upper and lower thresholds, while accuracy test compares the model, moving mean, moving median and previous sensor readings to the current value to determine accuracy, followed by a PCA test for the sensors that are deemed accurate in these tests.

Threshold accuracy is a pre-processing step to see if the recorded value is within given threshold limits. It is crucial, that the pre-processing steps are done before the actual accuracy monitoring to all sensor readings, to get data elements that have been checked and fixed to a form where the accuracy can be determined within the threshold limits utilizing the redundancy assumption between the sensors.

The moving median test is done to spot drifting sensor readings, and to make sure that the belief about the quality of the data from a single sensor remains consistent. This is upheld by the notion of including the pre-processing data flag values and the previous accuracy data flag value of each sensor reading into the equation of assessing current data quality. A test to compare the previous value to the current value is also included, to see if there is consistency between the value change from one point to another. The limit used for this check is 1.5% difference, which is a theoretical value, which can change according to the system and data under quality assessment. This reflects the inconsistency test for the values, which is here prioritized for the accuracy. The

inconsistency flag value will be recorded, but it is less demanding regarding the change value (5%).

The data quality monitoring for each sensor reading culminates to the accuracy quality determination. It is important, that some reliable information is gained about the trustworthiness of the data accuracy. Data accuracy is affected by all other dimensions, which then translates to believability of the sensor readings.

To make sure that the accuracy of the sensor readings is secured in a consistent and in a reliable way, accuracy monitoring is done in two-folded manner. The used methods focus on different aspects that affect accuracy, which gives more perspective of it, making the quality flag result more believable. Where one method may falter, the other one should manage to catch the problem, and give a matching data flag according to the perceived quality. The redundancy assumption is utilized in the monitoring, which makes the whole procedure susceptible for error, if no high-quality data readings are received.

The approach described in the previous paragraph utilizes methods to determine the accurate sensor readings among all the received ones, based on the difference between the current and previous reading values, moving median and difference to the trained and recursive model readings, equation (22). Encountering a sudden false reading is flagged and an estimated value is imputed. This imputation process relies heavily on the accuracy of the models and other methods, to get an accurate estimation of the current values. Condition-based methods are described by (Mathis and Thonhauser 2007; Klein and Lehner 2009; Fan and Geerts 2012).

$$\begin{aligned} &\textit{If accuracy}(t) = 1 \\ &QC(t) = \textit{imputation value,} \end{aligned} \tag{22}$$

where *accuracy*(*t*) is the given accuracy flag for the current received sensor reading,
Imputation value is the chosen value between the autoregressive model,
recursive autoregressive model and the moving median.

$$If\ accuracy(t\text{-}1) = 5\ or\ 3 \tag{23}$$

$$if\ (\frac{1}{k}\sum_{t=n-k}^{n} QC(t) - \frac{1}{k\text{-}1}\sum_{t=n-k+1}^{n} accurate\ values(t)) > mean\ threshold$$

$$Or\ \ \frac{QC(t) - QC(t\text{-}1)}{QC(t)} > change\ threshold,$$

$$Accuracy(t) = 1,$$

$$QC(t) = mean\ accurate\ values\ (t\text{-}1).$$

$$Else$$

$$Accuracy(t) = 5,$$

$$Accurate\ values(t) = QC(t),$$

Where k is the inspection window, (here 1),
accurate values are the variable, where all the reading values that get accuracy value of 5 are gathered, while *mean accurate values* is mean out of the accurate values,
mean threshold is the threshold for mean difference in this test (here 0.5),
change threshold is the threshold for the change between previous and current value (here 1.5%) and
t is the timepoint for *the data, t = current time, t-1 = previous data point etc.*

$$If\ accuracy(t\text{-}1) = 1 \tag{24}$$

$$if\ (\frac{1}{k}\sum_{t=n-k}^{n} QC(t) - \frac{1}{k\text{-}1}\sum_{t=n-k+1}^{n} accurate\ values(t)) > mean\ threshold \quad Or$$

$$\frac{QC(t) - QC(t-1)}{QC(t)} > change\ threshold,$$

$$Accuracy(t) = 1,$$

$$QC(t) = mean\ accurate\ values\ (t\text{-}1),$$

$$Else$$

Accuracy(t) = 5,

Accurate values(t) = QC(t),

where *k* is the inspection window, (here 5),

$$\begin{aligned} &\textit{If threshold accuracy}(t) = 1 \textit{ or accessibility}(t) = 1 \textit{ or timeliness}(t) = 1 \\ &\textit{or consistency}(t) = 1 \\ &\textit{Accuracy}(t) = 1, \\ &\textit{QC}(t) = \textit{median accurate values } (t\text{-}1), \end{aligned} \tag{25}$$

$$\textit{mean accurate values}(t) = \frac{\textit{accurate value}(1) + \ldots + \textit{accurate value}(n)}{n} \tag{26}$$

where *n* is the number of accurate values available at time t.

Principal component analysis (Section 4.3.1, p. 38) is used to validate accuracy, in the sense of swift changes in sensor readings. The aim is to spot sudden changes in the principal component analysis values to spot deviation from the assumed redundancy between the sensors, equations (17)–(19). This allows the quality check to be able to spot slow, drifting errors and the quicker spikes in the sensor readings, that do not follow the redundancy between the sensors, and thus they can be assumed as errors. The validation being done after the data has been already flagged and imputed prior to the PCA, makes it a double check to see if something was missed by the prior check. Arguments could be made for using just one check, but online operation does require high level of accuracy from the data quality assessment, and accurate quality flags are necessary.

If the previous accuracy value for the sensor reading was good, the current mean will be compared to a previous accurate mean, along with a comparation between the previous and current value, to see if drastic changes have happened, equations (23), (24) and (25). The accurate values are gathered into one variable, which is used in the imputation, as seen in equation (26). If no accurate values are found in the check, model

value is used as the imputation value. This action is a precursor to what happens in the PCA-analysis. If the previous accuracy value for the sensor reading was bad, its mean is compared to moving mean of accurate readings. This is done to detect drifting of the values. If the sensor reading is deemed accurate within the limits of moving median check, it will be utilized in the PCA-analysis.

The individual sensor reading scores are made into squared prediction errors, which allows for the deviation to be spotted as spikes. Threshold is set for the square prediction error to assess the deviation from the redundant mean, equation (20). If the threshold is broken by a sensor, it is given a 'bad' data quality flag. This of course assumes, that at least one of the sensors stands correct in its reading value, which is not false to assume if the number of sensors measuring the same thing, being redundant, is high enough for the inspected phenomena

The squared prediction error is given a threshold value, that dictates when the principal component analysis gives a bad quality flag for a data point, the threshold being $1*10^{-22}$ in this case. The threshold could be changed in the range of the system easily, and making it dynamic, depending on the current state of the system would also be something to consider, but here a constant threshold is used.

Believability

The last data quality dimension, which collects all the other data quality flag values into a certain value, based on which the data can be said to be usable or not, is believability. The individual data flag values in the pre-processing determine the believability value in a set order resulting in a score of 1, 3 or 5, as seen in equation (27) and in Table 1.

$$\begin{aligned} &If\ 1 \in \{accessibility(t),\ completeness(t),\ consistency(t), \ldots \\ &timeliness(t),\ threshold\ accuracy(t),\ accuracy(t)\} \\ &Believability(t) = 1, \end{aligned} \tag{27}$$

Elseif $3 \in \{completeness(t), accuracy(t)\}$ *or* $1 \in \{interpretability(t)\}$

$Believability(t) = 3,$

Else

$Believability(t) = 5,$

where $accessibility(t), \ldots, believability(t)$ are the data quality flags at time t.

Comparison between imputing methods and choosing the better between them is also available within the framework. Data flagging being a precursory action in data quality monitoring means that the data should be as representative of the received raw data and further the system state as possible. Application utilizing the monitored data, check quality flags to evaluate whether to use datapoints in their operations.

The believability flagging has some gimmicks to it, where the final believability value is affected differently by different data quality dimensions. If some data quality dimension does not affect the result of believability drastically, it does not lower the believability value all the way down to 1. Example of this case is interpretability, on which a data quality value of 1 only lowers the believability to 3, as the uninterpretable value is easily replaced if the source of the data is known. Another case where the believability value can be 3 is if the accuracy data flag value is 3, as accuracy is main data quality dimension.

In the end, believability data quality monitoring is done, and the believability data quality flag is given, which is the sum of all the previous data quality monitoring results. The data quality flags given at the pre-processing phase and then at the accuracy monitoring phase, determine the final flag in sequential manner. If the data quality flag given in a previous data quality dimension check or monitoring was bad, the good data quality flag in the current test cannot fix the overall bad data quality of the inspected datapoint, equation (27). Imputations are done, but one can never really know of the

true and accurate content of a point if it is not available in the first place. It must be estimated, which does not make for accurate data, but is assumably close to reality.

After the believability is determined for each of the different signals, the whole process will start again for new incoming data. There is case to be made for utilizing the past believability values in the determination of the new quality value flags. This is not necessarily wise however, as the believability flag only sets a base statement about the quality being good or bad. It can be scaled according to the requirements of the system but making a statement that a sensor reading is of no accuracy only because the name in the incoming data was of low quality, is a presumption that cannot be made without evidence. That is why determining the current accuracy of the sensor reading (equations (23)–(25)) values utilizes the individual data flag values of the pre-processing phase and the past accuracy flag value, not the believability value. The believability value being used to guide accuracy would steer all the accuracy value towards false, even if that might not be the case.

6 RESULTS AND DISCUSSION

6.1 Initialization

The whole time for signal 1, a loop repeated seven times, 1 is in Figure 27. This simulates a week's worth of variance in the temperature available for district heating, but the differences are exaggerated for testing purposes. There is noise present in the data, which are clearly visible in Figure 28. Each loop is different, with its own random ranges generated at the beginning of the loop. The main structure is maintained because it simplifies the simulation of the algorithm.

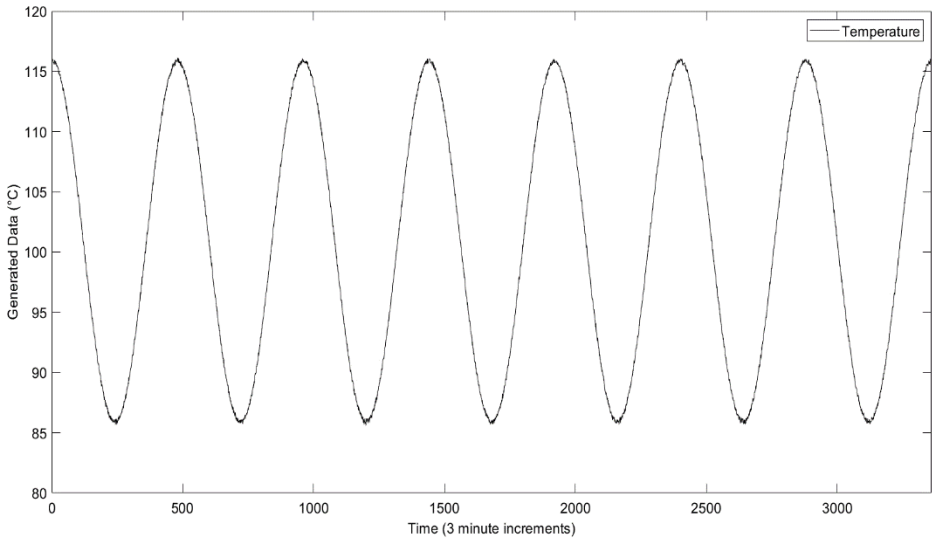

Figure 27. The whole range (seven days/seven loops) of the simulated temperature data (Signal 1).

The autoregressive models are trained for one loop iteration (one day), which then repeats itself seven times. It is important to have estimated value at an acceptable level of accuracy, which is harder to achieve when there is only one reading available, and it contains erroneous readings in it. To investigate the possible approaches, different for imputation values have been utilized, median out of moving median, recursive autoregressive- and autoregressive models, and their output is compared both with each other and the previous data point.

The results for the estimation of missing erroneous values for signal 1, with six sensor readings, can be seen in Figure 28. There are differences in the induced errors and how accurate the estimation is when the error exists. These results are from one iteration, for example, the last one (7^{th}) The results will be slightly different at each of the seven iterations, because of the randomization of the signal.

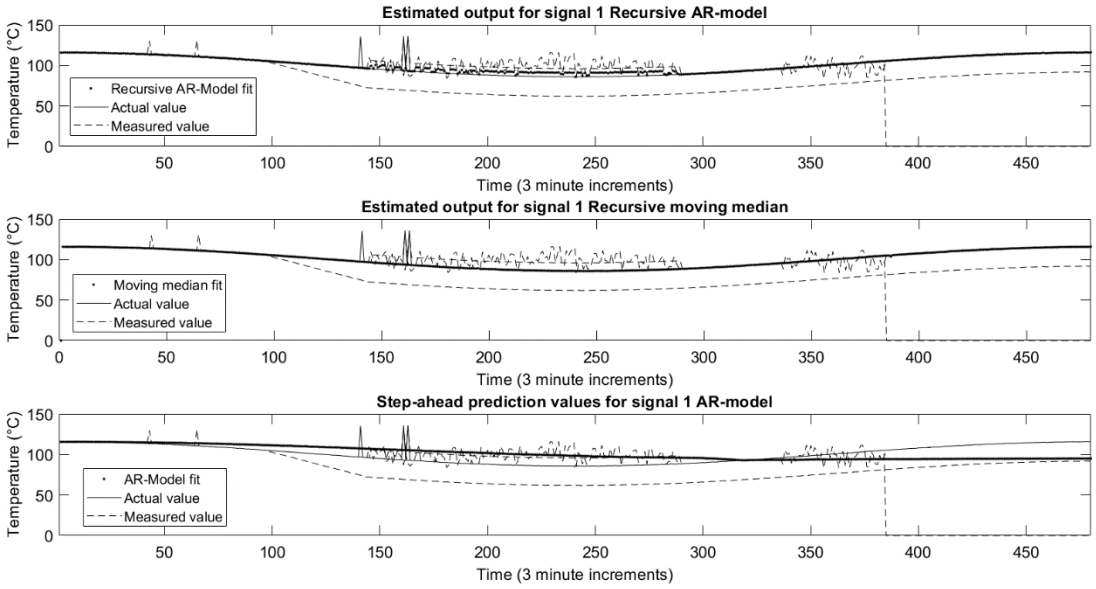

Figure 28. Signal 1 estimated values used for imputation, 6 sensor readings.

The used methods did not give accurate estimation results in the case of signal 2. The signal had drift error induced into it, leading all the values out of bounds, Figure 29. Other methods should be considered to deal with drift when only one sensor reading is available.

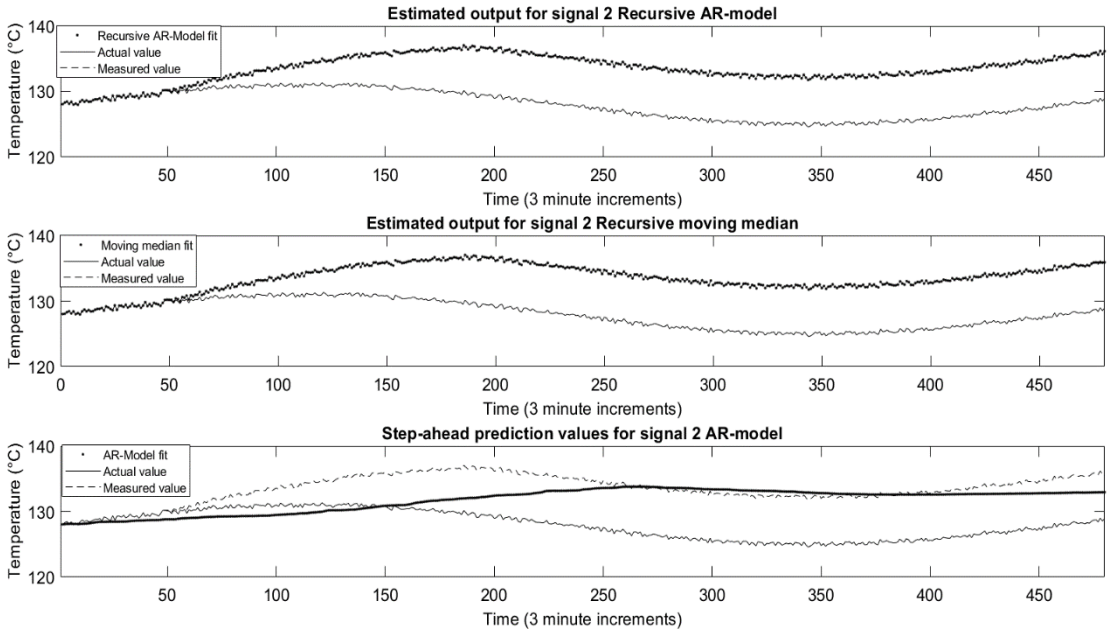

Figure 29. Signal 2 estimated values used for imputation.

Bias was induced into the sensor 3, which can be seen in Figure 30. Bias is seemingly much easier to handle than drifting, as Figure 29 shows compared to Figure 30 (Signal 2 compared to signal 3). The estimation was not ideal still, seeing there is a small section around increments 375–390, where none of the estimations are correct.

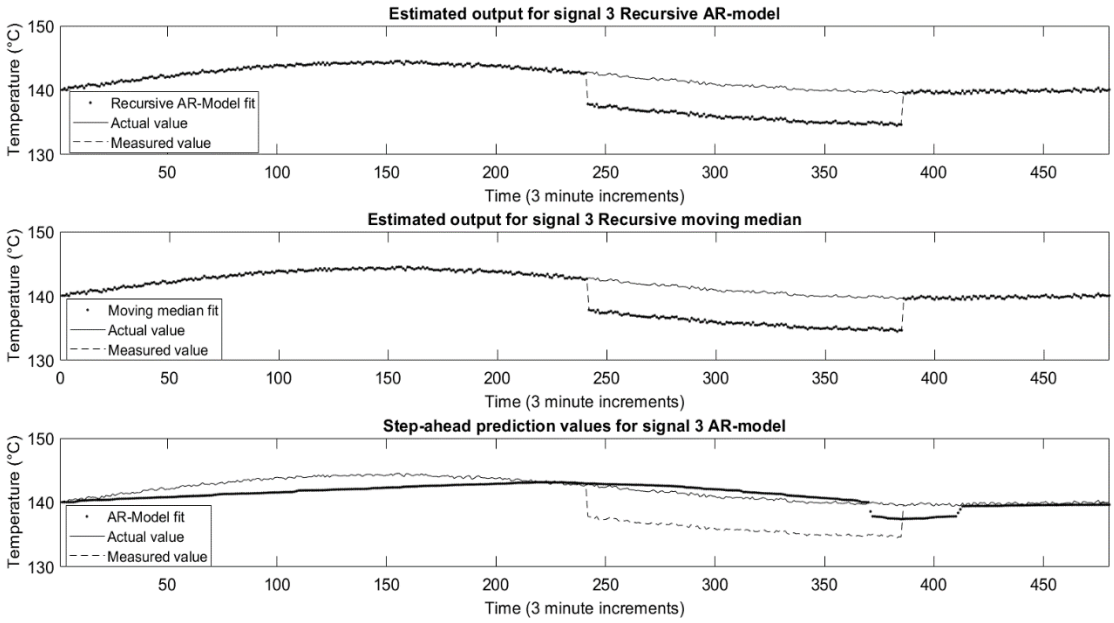

Figure 30. Signal 3 estimated values used for imputation.

The results seen in Figures 28–30 have noticeable deviation from the actual value. For example, signal 2 with the drifting results in estimations that share the trend but are of wrong magnitude. This results in bad accuracy values (quality flags of 1) for the signal 2, as the values are flagged low, starting from the point where the drift starts to take effect. In the signal 3, the different estimation methods can be seen to be having accurate estimation values in different sections of the signal length, which brings up the need for multiple different estimation methods, where the one that is deemed most accurate is utilized in the estimation. There is however a section, where all the estimation methods give erroneous estimation value, in which case the error cannot be fixed accurately by these methods.

The first signal (signal 1) can be seen to be having quite accurate estimation values from the moving median-method, which utilizes the redundancy between the sensor readings. The redundant values, even with error seem to give quite accurate estimation values, if at least one of the sensors is somewhat accurate. There are sections, when the algorithm is being run, where none of the measured sensors can be deemed accurate, in which case the imputation is done by the models. The section in the middle, where the recursive AR-model is having error, results in bad accuracy all around.

The problem with the imputation of missing or otherwise erroneous values boil down to the models being too inaccurate. Some expert data would be extremely useful even in this point of data quality flagging, which happens before the actual data is being utilized in the system. It would give some precious knowledge to choose the correct model-types and to understand when the system might be in an inaccurate state. With more resources and complex models utilized, better results could be achieved. That is, if it is required by the system.

6.2 Pre-processing

Interpretability deals with the metadata of a datapoint, meaning the time and name of the received value. The algorithm was able to successfully spot all 56 randomly imputed NaN values out of the metadata in the seven loops of the algorithm. Interpretability is the only data quality dimension, that effects accuracy and believability differently to the other pre-processing quality checks. A bad data quality flag from interpretability does

not affect the accurate sensor reading choice, as interpretability can be easily imputed when the data source is known, e.g., fixing the uninterpretable name with the previous name from the same source. Interpretability check results are seen below in Table 2. All the 56 occurrences of NaN were flagged.

Accessibility problems were simulated similarly to interpretability with random NaN values, but the accessibility check was done only to the value of the data element. This simulated the error, where the value could not be accessed and thus needed to be estimated by some means. If the value of the data was not accessible, the accessibility check would give a data quality flag value of 1. All 28 of these kinds of occurrences were spotted by the algorithm as shown in Table 2.

The completeness data quality check concerned all the elements in a datapoint, the value and the metadata, as it checked for missing values. These missing values were randomly distributed in the loop iteration, changing with each round, similarly to the NaN value distribution. Every element was checked independently in the completeness check, which made sure that all the ‘empty’-occurrences were spotted and then imputed with an estimated value, time, or name. As the estimation can never be trusted to be completely accurate, a special data flag of 9 was given to the imputation, resulting in the sensor believability data quality flag of 3. Not accurate but estimated to be close to the real value. All 84 empty data elements were spotted by the completeness data quality check as shown in Table 2.

Consistency check was done to evaluate the change between current and previous received data point values, with the threshold being 5% change. A change that exceeded this, was considered inconsistent, which meant a data quality flag of 1. This effected the determination of accurate sensor readings, as a sensor with an inconsistent change could not be considered accurate, thus resulting in an accuracy quality flag value of 1. There were no spotted changes in the metadata, but there was a total of 965 flagged occurrences of inconsistent change in the seven iterations out of 2847, as seen in table 2.

The threshold for timeliness, which gave ‘bad’ data quality flag of 1, was 2 seconds. The result of lowering the threshold to 1.5 seconds is in Table 2. The time was tied to the real-world time, which meant the algorithm had some time differences while going

through the loop, when more processing was required. Therefore, a timeliness threshold of 1 would mess up the whole loop and give out only bad quality flags, which in turn would mean that the accuracy of those elements would be 1, because if the data is not timely, it cannot be trusted to be accurate either. In a real-world application, where the difference in time of the data points would be longer, there would be no similar problems. A total of 222 occurrences where the difference between two datapoints was more than 1.5 seconds were spotted in the 7 iterations.

Given the number of forced and random errors induced into the signals, a lot of occurrences where the value went over the threshold took place, as seen in Figure 36. Not all these changes were caught 7737 out of 7743, which would lead one to believe, that the imputation caused the deviation. And the deviation would have to be ±5 in the signals 2 and 3 to exceed the thresholds, because of the inaccuracy of these models in the signals. Mostly in signal 2, as drifting caused a lot of issues for the value.

Table 2. Pre-processing quality flagging results.

Quality dimensions	Flagged occurrences (Value 1)	Total number of occurrences	Monitoring performance Correct flags, (%)
Accessibility	28	28	100
Interpretability	56	56	100
Completeness	84	84	100
Consistency	949	2847	33.33
Timeliness	222	222	100
Threshold accuracy	7737	7743	99.99

6.3 Accuracy and believability

In the accuracy determination, the first step was to ascertain the accurate sensor readings, and to check for a possible drift or sudden change in the values. This was done by the moving median check and by comparing the current and previous value difference with a tighter limit compared to the consistency value check namely 1.5%. These kinds of checks would not be accurate in a system where sudden changes are part of the normal operation conditions and different simple checks would be in place. However, in this case they work as sudden changes are not expected. These checks determine the most correct sensor readings, with the help of redundancy for the signal 1. The readings, that were considered erroneous here, are replaced by the ‘correct’ ones, which are determined by mean of the accurate readings, in the case of signal 1. Similar checks are done to the signals 2 and 3, without the added benefit of redundant values. The results for signals 1, 2 and 3 can be seen in, Figures 31–33.

If the pre-processing dimensions of accessibility, completeness, consistency, timeliness, or threshold accuracy had bad quality flag value (1) before this point, the accuracy was automatically deemed low, as in quality flag value of 1. This might be a cause for some of the spikes present in the quality managed data, but in this case, it is only for one point, so it did not remain, as the error was imputed for that point in time, Figure 31. The more lasting errors are due to inaccuracies in the values, which are then imputed to get an idea of accuracy when comparing all the redundant signal readings in sensors 1–6 to each other, as the incoming values can be compared to more accurate value to easier spot the deviation in the values.

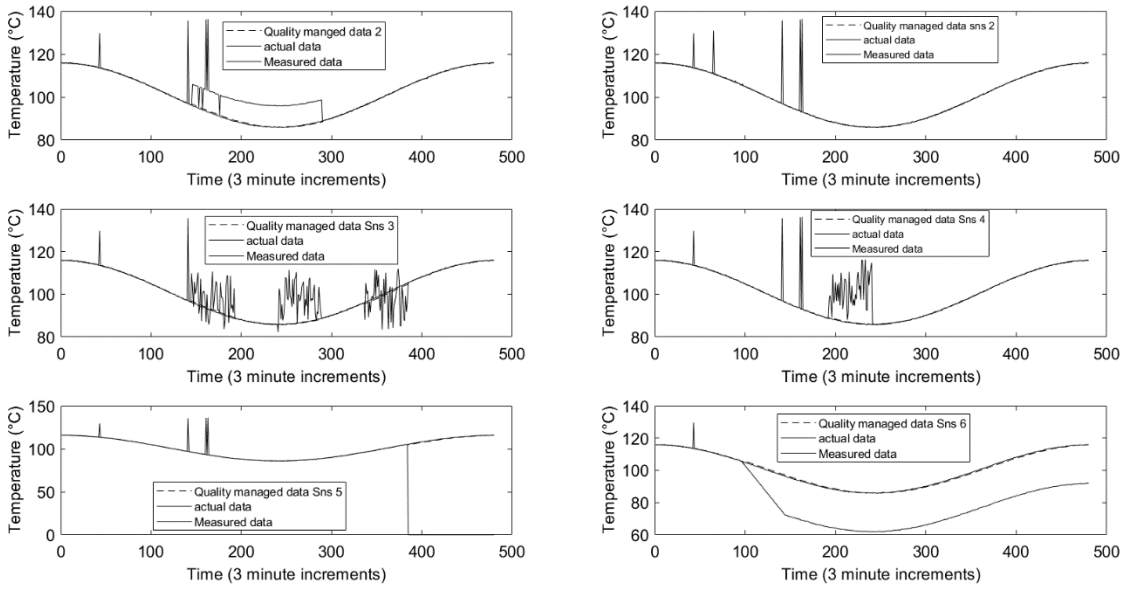

Figure 31. Sensor 1–6 values (Signal 1) after quality management.

There is still a lot of noise present especially in the signals 2 and 3 and some minor spikes in the values of the signal 1, with redundant values. The sensor 7 (signal 2) values after the quality management do not differ from the erroneous values almost at all, because the imputation methods are unable to estimate accurately due to the drifting error induced on the signal. The achieved quality is low, and the algorithm cannot deal with an error like this at its current state, Figure 32.

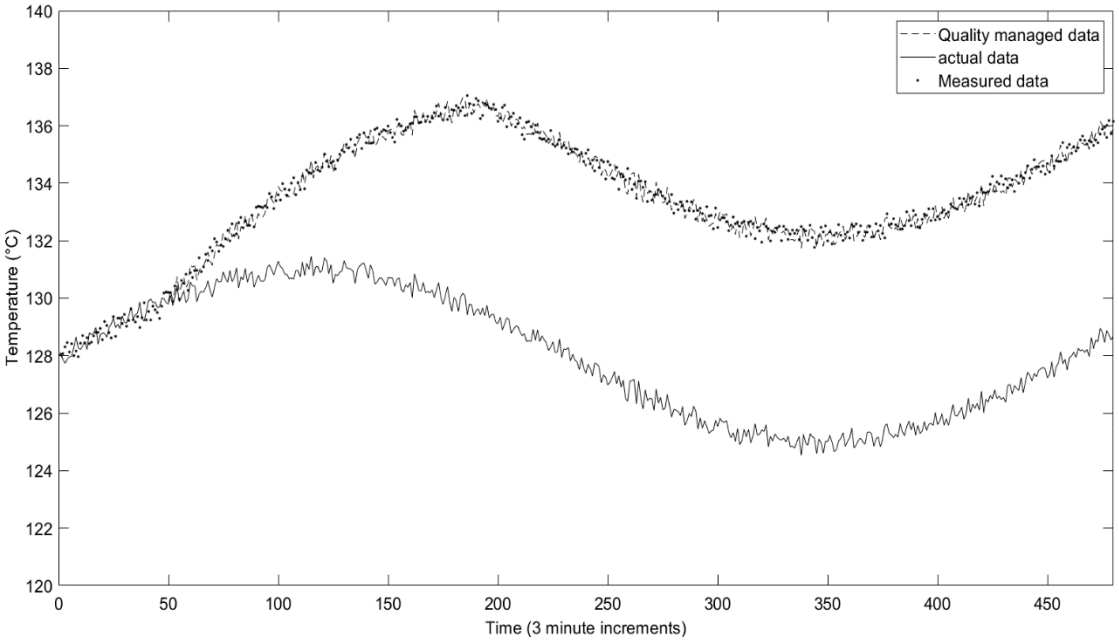

Figure 32. Sensor 7 values (Signal 2) after quality management.

The sensor 8 (signal 3) values can be seen in Figure 33. Between increments 373–391, where none of the estimation methods, the models or the moving median are sufficient, the value of the quality managed signal 3 can be seen dropping. The deviation from the actual value starts at increment 242, where the forced error starts taking effect. This results in bad accuracy rating for it, either by threshold breach or by moving median change or by change in the previous values.

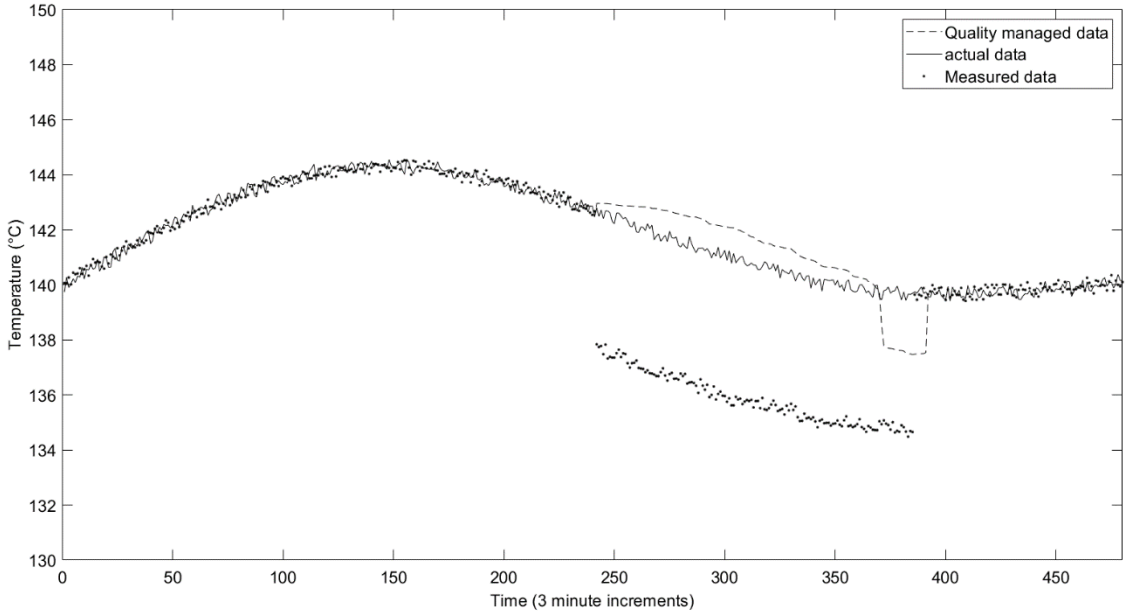

Figure 33. Sensor 8 values (Signal 3) after quality management.

These sensor readings of signal 1 were fed into the principal component analysis, based upon the previous test of whether the sensor reading was deemed accurate or not. The number of accurate sensors in each iteration is shown in Figure 34. The difference between the number of accurate sensors is noticeable in the 125–300 data range of the seventh loop data. The general number of accurate sensors seems to follow the trend of forced errors accurately, while there are some observable deviations caused by the random errors. The difference between accurate sensor count is shown in Figure 34.

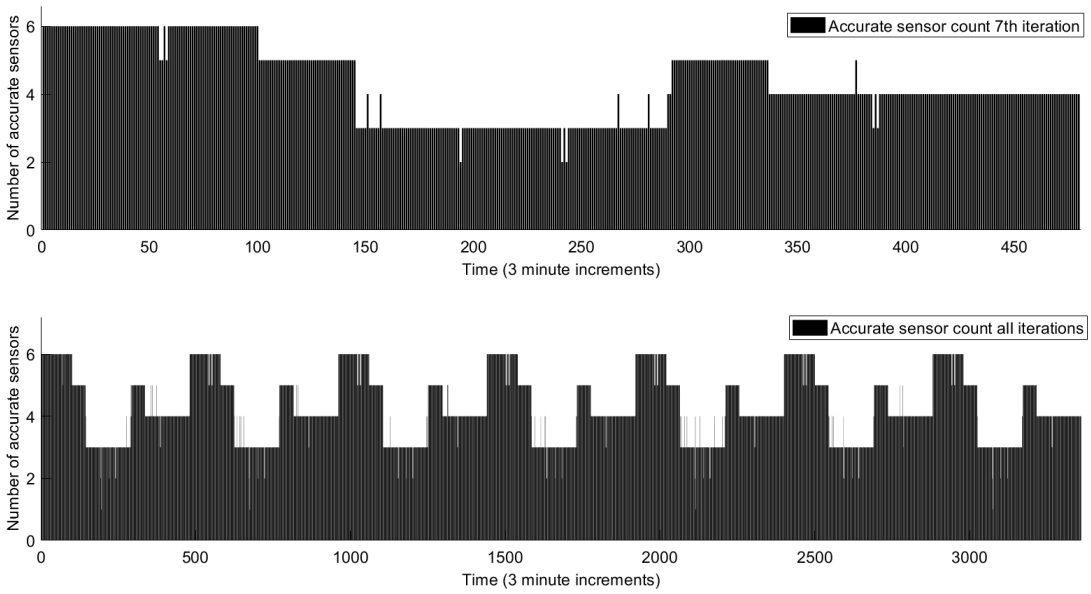

Figure 34. Number of accurate redundant sensor readings before PCA in signal 1, above 7th loop iteration, below all loop iterations.

In the principal component analysis phase, the first principal component was used. The explained values derived from that first component are presented for one and for all rounds in Figure 35. There is a noticeable trend with some "random" variation, which can be attributed to the forced error that happened in known ranges of one round. The random variation could be explained by the random errors. The lowered explanation values could lead to bad estimations, which would make the case for more principal components being used, but it is a case to be inspected in future studies. The squared prediction errors gained from using the first principal components were the basis for giving bad or good quality flags, to which the explanation value can influence. If it is too low, the estimation will be erroneous leading into false bad quality flags.

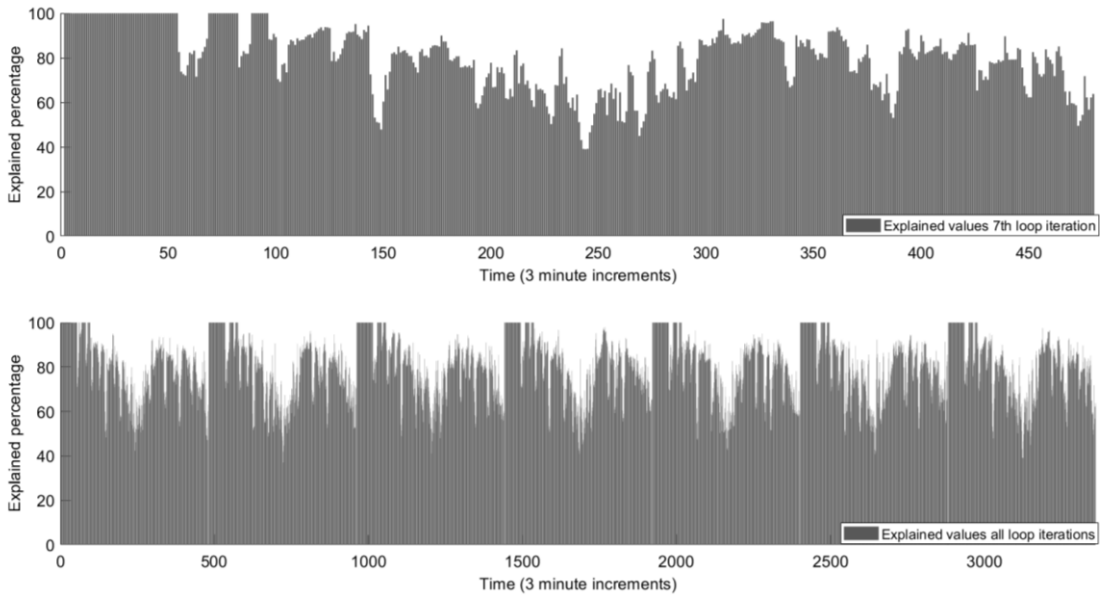

Figure 35. Explained values from principal component analysis, above seventh loop iteration and below all loop iterations.

The squared prediction errors tell of the deviation from the expected values. As mentioned, earlier, the low explained values and the randomness present in the data might explain why sometimes after the accuracy checking phase, all squared prediction errors of the sensors deemed accurate go over the threshold (Figure 36). This is an error that could be fixed by including more principal components into the analysis. One principal component seems to be sufficient to a certain extent, but with the random noise and sudden changes in the data can lead to the principal component analysis being impractical. The points that had all the ‘accurate’ sensor readings going over the threshold were given a quality flag of 3, as it is most probable, that they are due to the reasons explained before.

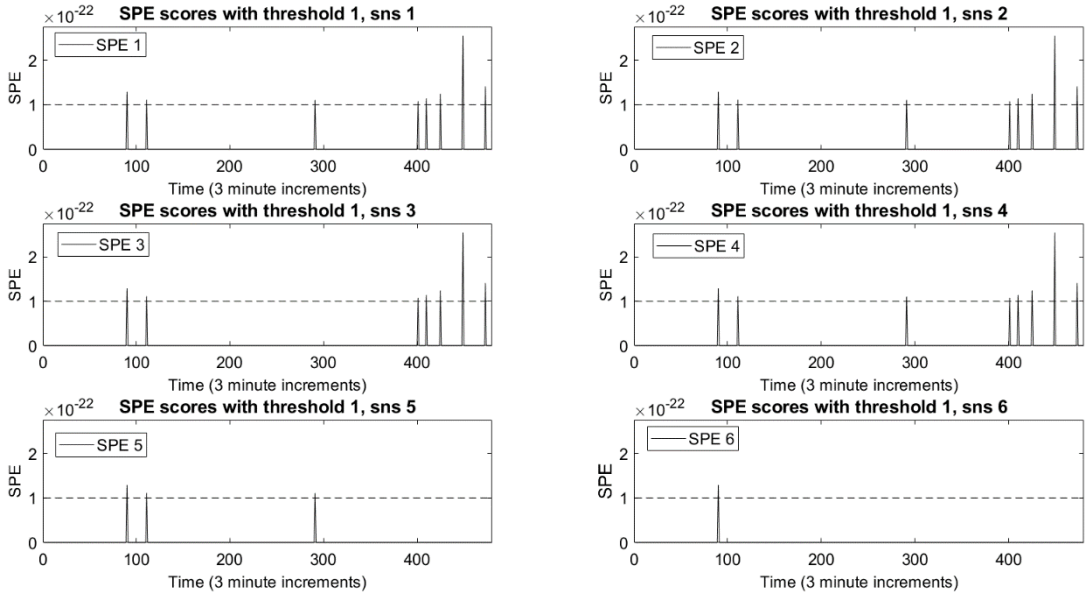

Figure 36. SPE-values over the threshold in the 'accurate' sensors, only threshold breaches, seventh loop iteration.

Looking at the SPE values of all loops, a trend is found especially in the sensor 2, 3 and 4 (Figure 37). It seems that the sensor squared prediction errors mimic each other, after the forced errors take effect and the imputations start happening. This is most probably due to the values in the sensors being so similar (mean difference of $0.182*10^{-25}$, while the scale of the SPE is 10^{-22}), due to the redundancy. The effect of the PCA does seem underwhelming, but with some tweaking of the threshold value and including more principal components, this would probably be more accurate to represent sudden deviation from the redundant values.

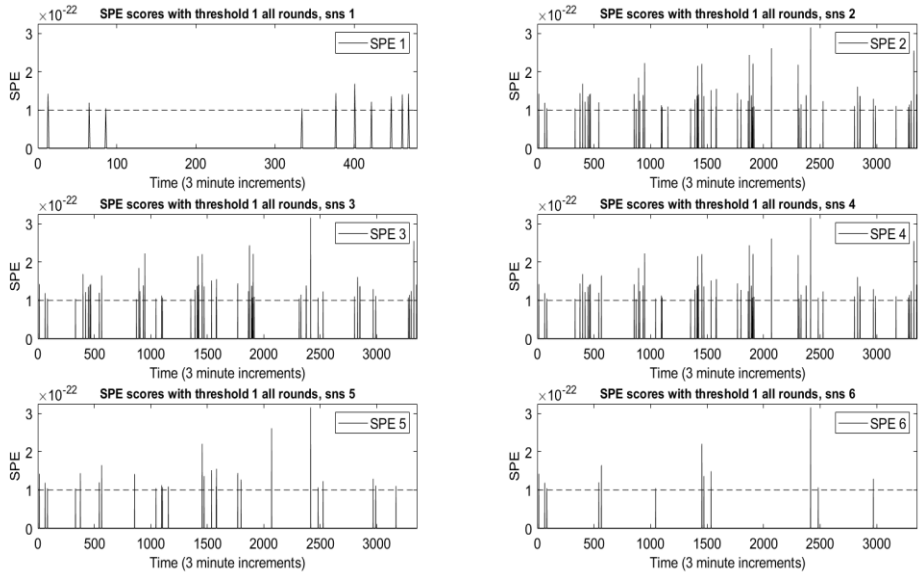

Figure 37. SPE-values over the threshold in the 'accurate' sensors, only threshold breaches.

The amount of quality flags given at PCA is around 2% of the total amount of low or mediocre quality data flags. This seems to indicate, that the accuracy checks per sensor done before the PCA is sufficient in fixing the accuracy in the simple simulation case of this study. In the future, changing the threshold values used in the study could yield some different results, giving the PCA more weight, but this does not seem to be necessary, at least in this case. The method would require some further validation in cases, where two-fold data accuracy validation in a series would be required to combat rapid changes. Nonetheless, the PCA seems to be working as intended in respect to the quality flags.

There was a total of 8 sensors, each of them getting 7*480 = 3360 sensor readings, totalling 8*3360 = 26880. The believability flags were divided into 3 categories. The 'Good' flags with the value of 5 was given to a total of 15485, 'Mediocre' flags with the value of 3 was given a total to 339 and 'Bad' believability quality flags with the value of 1 was given to a total of 11056 sensor readings. There is a sharp difference between the number of data quality flags of 5 to 3 and then again from data quality flag 3 amount to data quality flag value of 1. This is simply a scaling phenomenon, which can be done in any which way the user wants to. The large number of believability flags of 1 is most

probably due to the signals 2 and 3, as they have a lot of imputed error values. The algorithm works as intended, by flagging the erroneous sensor readings as bad and letting the users know that the data is inaccurate. Data flagging is meant to do just that, nothing more. Handling data that consists of only bad quality flags is what starts to be a problem. It is the natural next step after this data flagging phase, which was addressed to some extent with the first signal with redundant reading values.

6.4 Practical implications and future aspects

In the context of this thesis, the area of study is limited to the received data by the system, and the integrity of the data source is not assessed. Then, validity of the data source is assumed to be good, or it is assumed that the data source integrity has been assessed before this quality monitoring phase. The data monitoring and flagging is just a piece of the bigger puzzle that is data quality control, which can be a large subject area, depending on the intended use.

The algorithm presented in this thesis is built in a way, that it must impute missing values and replace uninterpretable elements, so it can assess and give a quality flag for accuracy and believability. This creates a need to estimate the values or elements if they are missing or uninterpretable, which can be challenging in a real-world scenario. However, many of these methods are already in use, so it should be a realistic case.

The way accuracy is flagged in this algorithm is an approach that was developed just for this case study. It still has potential to become effective way of monitoring the data quality in a way that considers the future, present and the past, taking more things into consideration, than present data quality monitoring approaches. The fact that wrong values will most probably keep producing wrong values is built into this algorithm, with exhaustive checks to go through if bad accuracy is encountered. Real-time property makes the proposed data quality monitoring concept a promising one.

The accuracy of the recursive autoregressive and the autoregressive model are important for estimating the erroneous values. Data that is continuously erroneous over a period, will need accurate estimations. The online estimation of the current value in absence of

value for the sensor cannot be too complex, or it will be too slow. The methods used here are just a careful approach to the topic and in no sense best possible ones.

So many missed quality flags in the consistency quality check remains unsolved. It might just be that it is due to the method the inconsistent changes were calculated. But it is something that tells this algorithm is not done, far from it. Giving only 33% of the bad quality flags that should be given is not tolerable.

Considering the data is created, separated into multiple inspected points, and then assessed for quality, it is computationally heavy to process. An option to dodge this issue would be to divide the data creation, data error simulation and the data quality assessment into their own separate denominations, which would allow for much smoother runtimes as it would be potentially easier to process. This would mirror real-life data quality assessment, as it is a separate function to the process creating the data.

This theorical setting is ideal for this kind of assessment, but real-life scenarios would require extended efforts to figure out where redundancy is present among the numerous sensors, if any is present at all. This brings up the question of using other approaches, or maybe combinations of approaches, but that will be delved into in future research. Approaches that are based on statistical methods come to mind, as talked in the theory section.

The choice of utilizing PCA to confirm the redundancy and to further improving the quality flagging accuracy was an experimental one. The amount of quality flags given at the PCA-phase ended up being low in this simulation. The quality assessment done before it seemed to do most of the work. Assessing the final impact of the PCA-analysis could be done from the quality flags given at that point, compared to all the accuracy quality flags. The impact would be around 2%, out of which most might be unreliable due to the random errors and low explanation values. Other variables from the principal component analysis could also be utilized, like Hotelling’s t-squared statistic, which would indicate deviation from the expected result and SVI (sensor variable index), which would indicate the sensor with the wrong value. These methods could increase the benefit gained from PCA.

When all the squared predicted errors in the sensor readings that have been determined accurate in the moving median check are over the threshold, the accuracy value is determined to be 3. This assumes a lot about the systems behaviour, because it is most likely caused by noise, but it can also be wrong. In the future this should be more closely inspected. The redundancy feature gives an error in this situation, as all the values are usually the same and go over the threshold, so there is little information to diagnose the cause of this error. It might be due to the heavy noise imputed into the signals. Or it might be due to the low explained-value from the PCA-analysis in the inspected point and would require more principal components to be used. This is something that would need to be monitored with a system that has access to historical data, making the tuning the algorithm easier.

The redundancy assumption would be a question when moving the algorithm to real-life applications. How to determine what sensors have redundant values and basically define the limits for this kind of approach? Some methods, like centring could be utilized to take advantage of sensors that measure similar trend of values, but with different amplitudes, or perhaps scaling or weighing the different readings depending on the application. Centrepoint could be chosen, to which all the other sensor readings would mirror themselves to, to utilize the wide range of sensor readings in a real-life application. Change in a measured value could indicate that the later measuring values should change as well, at least to some extent, which would make the case for delay or staggering the quality analysis.

When imputation is done with the median of the moving medians, redundant sensor readings could potentially have some use. It is simple but does require the simulation of redundant signals with forced error. To some extent this is a statistical estimation method where all the potential errors are considered. The actual sensor reading should mirror the one simulated redundant signal with similar error, and thus push the estimation towards it, unless the sensor values are noticeably wrong, in which case a trained model estimation would be used. Some historical and expert knowledge would aid in utilizing this method.

One way for estimating the real value would thus be measuring the actual value from a single sensor, and then simulating possible error that could take place, making

numerous redundant sensors reading values, that can be utilized with the median out of the moving medians of all the simulated sensor readings together with the actual one. This would have a similar approach to the estimation as model predictive control, but it would be little more simplified, and maybe require less steps to be taken depending on the required accuracy.

Acknowledging the fact that data quality determination is all based on the way it is defined, picking as many quality dimensions as possible would make for most accurate description. While this is true, the errors in other less important dimensions, can hide errors in the more important dimensions, which are more important for the data quality in general. This might cause the system to avoid repairing some issue it faces, because it spotted only the minor error. Avoiding this from happening could be built into the algorithm, but it needs to be acknowledged, nevertheless.

Change in the quantity of data would also demonstrate the accuracy of the algorithm in action. Between 85% and 95% streaming error identification accuracy is seen in the algorithms tested by Luo *et al*. (2019). To test this similarly with this data would require similar benchmark data, that could show difference between different algorithms. The chosen data should be energy related still, due to the format differences in the data sets.

The problems, faced in the data estimation, boil down to the bad estimation of the true value, that is being measured. Better estimation, in real-time application, would fix the problems faced in this study with simple estimation methods. Choosing the best method for online estimation of the true values being measured would be the next step after this study along with other optimization actions. Doing that would also require closer inspection on the chosen inspection ranges, and threshold values, that determine when the bad quality flag is given.

To improve the algorithm, there is much to be optimized. Including methods to estimate and evaluate the signal value validity together with making the algorithm easier to use. Some aspects of predicting the system behaviour could and should be added into the algorithm to make it more robust, thus making dealing with errors easier. Predicting multiple different scenarios, where the system could go wrong, would be a good way to do approach this problem together with validating the correctness of data.

What the algorithm can achieve with the simulated data is at an acceptable level. The algorithm needs to be tested with real-life systems and with real-life data, to adjust it to work with that kind of system or data and to see how the chosen methods perform with real data and to validate algorithm's performance. The online aspect of it is what makes optimizing it challenging, but with current processing power, that should not be a problem. The memory used before the plotting was around 611 MB.

What would be required from the system, in the sense of data and metadata, should at least be what is included in the algorithm here. The value, the creation time of the datapoint, and the name of the data source. More available metadata would give more options for data quality assessment and better pre-processing options, but it would also make the data quality assessment and data quality flagging more complex, as it is convoluted as it is.

The accuracy limits of the measured temperature represented in this study revolve around 1–1.5%, but the value would most likely change depending on what the error percentage is based upon, for example unit of temperature, like in the case of this study. From a data standpoint, accurate data is always better, as it requires less managing. Errors are however inevitable, which is why a certain perceived range of accuracy from the system is required and assumed in the data analysis. This algorithm bases the perceived accuracy of the values upon the received sensor values, comparing them with different methods of estimation. However, this kind of approach would fail if the received data was completely oblivious to the real system state.

7 CONCLUSIONS

This algorithm does accuracy monitoring in series and takes into consideration the previous data flags. It is a novel way to approach data quality monitoring task, as an error usually does not come alone, it keeps being bad. The loop back function in the algorithm is also necessary and helpful to steer the quality flagging to more accurate flags, as only the data that is deemed accurate is considered in the estimation. This turned out to be accurate way to estimate the actual output, with only some simple methods set in series within a loop.

Data quality monitoring is a complex matter, but this study shows that it can be divided into subsections which are easier to distinguish. This allows inspection of individual aspects related to the data quality and thus eases the process of data quality monitoring and flagging. The data quality dimensions must be carefully specified and selected, as the data quality monitoring system will be built around them. Otherwise, the system will fail to find all the data quality deficiencies that would be necessary to find out.

What is required to be in the data is at least the value, time of capture and name of the sensor. More information about the data would allow for more specific inspection of the quality and dependencies between different elements. Basic inspection can be done only with the three described data elements.

The algorithm made for this case monitors and flags all the deficient data it encounters. However, there are certain issues related to the inaccurate estimations, namely when the estimate of the data must be imputed to replace the erroneous data point. If the estimation is inaccurate and there is lacking information about the system state, the accurate state might be assumed to be wrong, and the data quality monitoring method might give biased flags.

The data window range influences the result, naturally changing the outcome of the monitoring system. By tightening the data ranges and limits, the given data flags would accurately describe the system, but would also possibly affect its operation. The threshold values must be therefore chosen on the spot to match the system operation requirements.

8 SUMMARY

Data quality is a complex area, more so with the increasing volume and complexity of data. How data quality is perceived boils down to how it is defined, and how accurately the different quality dimensions of data are determined. This is also dependent on what the data is described by the system that produces it, in other words how accurate the metadata is. The performance of an individual algorithm is thus dependent on the data it has been built to assess.

In summary, the algorithm does what it was designed to do, and it spots lacking quality in the incoming data giving data flags. The next natural step would be to test the function of the algorithm with different data and see how it performs, when the thresholds and ranges are adjusted for a different kind of situation. Different situations would call for different methods, which would need different estimation methods for the data, which poses as a problem, because not all estimation methods work for all situations.

The way the accuracy is determined within the algorithm, considering past, present, and future values, and having two quality checks in a series is a new way to approach the issue of data quality assurance. It was developed for the purpose of guiding the data quality flags exhaustively to the correct values, while considering all the quality dimensions in real-time operation. This limits the range of methods that can be utilized within these constraints.

The estimation of the actual values needs to be chosen differently if no redundant sensor readings are simulated. The methods of this thesis are based on simple range and value checks mostly together with PCA to determine the accuracy, but more complex methods based on statistics for example could be utilized to better describe the data quality through the quality flags. The general accuracy of existing algorithms that detect erroneous data in online operation is around 85–95%, which this algorithm should be tested against with different data.

REFERENCES

Acuna, E. and Rodriguez, C., 2004. Classification, Clustering, and Data Mining Applications. Classification, Clustering, and Data Mining Applications, (June), 639–647.

Alahakoon, D. and Yu, X., 2016. Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey. IEEE Transactions on Industrial Informatics, 12 (1), 425–436.

Allalouf, M., Gershinsky, G., Lewin-Eytan, L., and Naor, J., 2014. Smart grid network optimization: Data-quality-aware volume reduction. IEEE Systems Journal, 8 (2), 450–460.

Ancillotti, E., Bruno, R., and Conti, M., 2013. The role of communication systems in smart grids: Architectures, technical solutions and research challenges. Computer Communications [online], 36 (17–18), 1665–1697. Available from: http://dx.doi.org/10.1016/j.comcom.2013.09.004.

Ardagna, D., Cappiello, C., Samá, W., and Vitali, M., 2018. Context-aware data quality assessment for big data. Future Generation Computer Systems [online], 89, 548–562. Available from: https://doi.org/10.1016/j.future.2018.07.014.

Ballabio, D., 2015. A MATLAB toolbox for Principal Component Analysis and unsupervised exploration of data structure. Chemometrics and Intelligent Laboratory Systems [online], 149, 1–9. Available from: http://dx.doi.org/10.1016/j.chemolab.2015.10.003.

Baskar, S., Arockiam, L., and Charles, S., 2013. A Systematic Approach on Data Pre-processing In Data Mining. Compusoft [online], 2 (11), 335–339. Available from: http://journaldatabase.info/articles/systematic_approach_on_data.html.

Batini, C., 2016. Data and Information Quality - Dimensions, Principles and Techniques. Milan: Springer International.

Bello-Orgaz, G., Jung, J. J., and Camacho, D., 2016. Social big data: Recent achievements and new challenges. Information Fusion [online], 28, 45–59. Available from: http://dx.doi.org/10.1016/j.inffus.2015.08.005.

Bhattarai, B. P., Paudyal, S., Luo, Y., Mohanpurkar, M., Cheung, K., Tonkoski, R., Hovsapian, R., Myers, K. S., Zhang, R., Zhao, P., Manic, M., Zhang, S., and Zhang, X., 2019. Big data analytics in smart grids: State-of-theart, challenges, opportunities, and future directions. IET Smart Grid, 2 (2), 141–154.

Bizer, C. and Cyganiak, R., 2009. Quality-driven information filtering using the WIQA policy framework. Web Semantics, 7 (1), 1–10.

Bro, R. and Smilde, A. K., 2014. Principal component analysis. Analytical Methods, 6 (9), 2812–2831.

Catterson, V. M. and McArthur, S. D. J., 2016. Data Analytics for Transmission and Distribution. Smart Grid Handbook, 1–19.

Chen, H., Qiu, M., Ge, H., and Li, M., 2017. The Application of Energy Network Theory in the Analysis of District Electricity and Heating System. Proceedings - 1st IEEE International Conference on Energy Internet*,* ICEI *2017*, 36–41.

Chen, H., Qiu, M., and Ngan, H. W., 2019. Energy Network Theory for Modeling and Analysis of Integrated Energy Systems. 2018 International Conference on Power System Technology, POWERCON 2018 - Proceedings, (201804270001126), 424–432.

Chen, W., Zhou, K., Yang, S., and Wu, C., 2017. Data quality of electricity consumption data in a smart grid environment. Renewable and Sustainable Energy Reviews [online], 75 (October 2016), 98–105. Available from: http://dx.doi.org/10.1016/j.rser.2016.10.054.

Cheng, L., Zhang, Z., Jiang, H., and Yu, T., 2018. Local Energy Management and Optimization : A Novel Energy Universal Service Bus System.

Diamantoulakis, P. D., Kapinas, V. M., and Karagiannidis, G. K., 2015. Big Data Analytics for Dynamic Energy Management in Smart Grids. Big Data Research [online], 2 (3), 94–101. Available from: http://dx.doi.org/10.1016/j.bdr.2015.03.003.

Ehrlinger, L., Werth, B., and Wöß, W., 2018a. QuaIIe: a data quality assessment tool for integrated information systems. Proceedings of the Tenth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2018), (May), 21–31.

Ehrlinger, L., Werth, B., and Wöß, W., 2018b. Automated Continuous Data Quality Measurement with QuaIIe. International Journal on Advances in Software, 11 (3 & 4), 400–417.

Even, A. and Shankaranarayanan, G., 2005. Value-driven data quality assessment. Proceedings of the 2005 International Conference on Information Quality, ICIQ 2005.

Fan, W. and Geerts, F., 2012. SYNTHESIS LECTURES ON DATA MANAGEMENT Foundations of Data Quality Management. Morgan & Claypool publishers [online], 219. Available from: https://www-morganclaypool-com.libproxy.ucl.ac.uk/doi/pdf/10.2200/S00439ED1V01Y201207DTM030.

Feeney, K. C., O'Sullivan, D., Tai, W., and Brennan, R., 2014. Improving curated web-data quality with structured harvesting and assessment. International Journal on Semantic Web and Information Systems, 10 (2), 35–62.

Flemming, A., Freytag, J., and Paschke, A., 2011. Qualitätsmerkmale von Linked Data-veröffentlichenden Datenquellen. Recherche, 1–174.

Fürber, C. and Hepp, M., 2011. SWIQA - A Semantic Web information quality assessment framework. 19th European Conference on Information Systems, ECIS 2011.

García, S., Luengo, J., and Herrera, F., 2015. Data Preprocessing in Data Mining. 72nd

ed. Intelligent Systems Reference Library. Granada: Cham: Springer. 2014.

Ge, M., Chren, S., Rossi, B., and Pitner, T., 2019. Data Quality Management Framework for Smart Grid Systems. In: Abramowicz, W. and Corchuelo, R., eds. Business Information Systems. Cham: Springer International Publishing, 299–310.

Gu, C., Groth, P., Stadler, C., and Lehmann, J., 2012. Using Network Measures, 87–102.

Gürcan, Ö. F. and Yazici, İ., 2017. Big data and energy - A review. In: International Symposium on Industry 4.0 and Applications (ISIA 2017) [online]. Karabük, 86–91. Available from: https://www.researchgate.net/publication/322570871_Big_data_and_energy-A_review.

Han, J., Kamber, M., and Pei, J., 2012. Data Mining: Concepts and Techniques. Data Mining: Concepts and Techniques. Cambridge: Cambridge University Press.

Hartig, O., 2008. Trustworthiness of Data on the Web. Science [online], 9, 1–5. Available from: http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/conferences/Hartig_STIPhDWorkshop.pdf.

Hazen, B. T., Boone, C. A., Ezell, J. D., and Jones-Farmer, L. A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. International Journal of Production Economics [online], 154, 72–80. Available from: http://dx.doi.org/10.1016/j.ijpe.2014.04.018.

Hou, W., Ning, Z., Guo, L., and Zhang, X., 2019. Temporal, functional and spatial big data computing framework for large-scale smart grid. IEEE Transactions on Emerging Topics in Computing, 7 (3), 369–379.

Jaradat, M., Jarrah, M., Bousselham, A., Jararweh, Y., and Al-Ayyoub, M., 2015. The internet of energy: Smart sensor networks and big data management for smart grid. Procedia Computer Science [online], 56 (1), 592–597. Available from:

http://dx.doi.org/10.1016/j.procs.2015.07.250.

Kantardzic, M., 2011. Data Mining: Concepts, Models, Methods, and Algorithms: Second Edition. Data Mining: Concepts, Models, Methods, and Algorithms: Second Edition.

Klein, A. and Lehner, W., 2009. Representing data quality in sensor data streaming environments. Journal of Data and Information Quality, 1 (2).

Kontokostas, D., Westphal, P., Cornelissen, R., Bibliotheek, S., Hellmann, S., and Lehmann, J., 2014. Test-driven Evaluation of Linked Data Quality Categories and Subject Descriptors. Www2014, 747–757.

Koseleva, N. and Ropaite, G., 2017. Big Data in Building Energy Efficiency: Understanding of Big Data and Main Challenges. Procedia Engineering [online], 172, 544–549. Available from: http://dx.doi.org/10.1016/j.proeng.2017.02.064.

Krishnan, S., Haas, D., Franklin, M. J., and Wu, E., 2016. Towards reliable interactive data cleaning: A user survey and recommendations. HILDA 2016 - Proceedings of the Workshop on Human-In-the-Loop Data Analytics, (1), 1–5.

Lee, I., 2017. Big data: Dimensions, evolution, impacts, and challenges. Business Horizons [online], 60 (3), 293–303. Available from: http://dx.doi.org/10.1016/j.bushor.2017.01.004.

Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y., 2002. AIMQ: A methodology for information quality assessment. Information and Management, 40 (2), 133–146.

Li, B., Roche, R., Paire, D., and Miraoui, A., 2018. Coordinated scheduling of a gas/electricity/heat supply network considering temporal-spatial electric vehicle demands. Electric Power Systems Research [online], 163 (July), 382–395. Available from: https://doi.org/10.1016/j.epsr.2018.07.014.

Liu, X., Wang, S., and Sun, J., 2018. Energy management for community energy

network with CHP based on cooperative game. Energies, 11 (5).

Loshin, D., 2011a. Business Impacts of Poor Data Quality. The Practitioner's Guide to Data Quality Improvement, 1–16.

Loshin, D., 2011b. Dimensions of Data Quality. The Practitioner's Guide to Data Quality Improvement, 129–146.

Loshin, D., 2011c. Metadata and Data Standards. The Practitioner's Guide to Data Quality Improvement, 167–189.

Loshin, D., 2011d. Data Requirements Analysis. The Practitioner's Guide to Data Quality Improvement, 147–165.

Luo, H., Sun, K., Wang, J., Liu, C., Ding, L., and Song, B., 2019. Multistage identification method for real-time abnormal events of streaming data. International Journal of Distributed Sensor Networks, 15 (12).

Mathis, W. and Thonhauser, G., 2007. Mastering real-time data quality control-how to measure and manage the quality of (rig) sensor data. Proceedings of the SPE/IADC Middle East Drilling Technology Conference and Exhibition, 172–181.

Mayer-Schönberger, V. and Cukier, K., 2013. Big Data: A Revolution That Will Transform How We Live, Work, and Think. International Journal of Advertising.

Mendel, J. M. and Korjani, M. M., 2014. On establishing nonlinear combinations of variables from small to big data for use in later processing. Information Sciences [online], 280, 98–110. Available from: http://dx.doi.org/10.1016/j.ins.2014.04.042.

Michael, P., 2015. Continuous Data Quality Assessment in Information Systems. Journal of Hydroinformatics [online], 17 (4), 144. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84936748281&doi=10.2166%2Fhydro.2015.042&partnerID=40&md5=07b085e10a90f6d71973e52090296fb5.

Moossavizadeh, S. M. H., Mohsenzadeh, M., and Arshadi, N., 2012. A new approach to measure believability dimension of data quality. Management Science Letters, 2 (7), 2565–2570.

Moreira, J. M., de Carvalho, A. C. P. L. F., and Horváth, T., 2018. *A General* Introduction to Data Analytics. A General Introduction to Data Analytics.

Olson, J. E., 2003. Data Quality: The Accuracy Dimension. Data Quality: The Accuracy Dimension. San Francisco: San Francisco : Morgan Kaufmann (The Morgan Kaufmann Series in Data Management Systems).

Paulheim, H. and Bizer, C., 2014. Improving the quality of linked data using statistical distributions. International Journal on Semantic Web and Information Systems, 10 (2), 63–86.

Phan, S. K. and Chen, C., 2017. Big Data and Monitoring the Grid [online]. The Power Grid: Smart, Secure, Green and Reliable. Elsevier Ltd. Available from: http://dx.doi.org/10.1016/B978-0-12-805321-8.00009-4.

Pipino, L. L., Lee, Y. W., and Wang, R. Y., 2002. Data Quality Assessment. Communications of the ACM, 45 (4), 211–218.

Pradhan, S., 2005. Believability as an information quality dimension. Proceedings of the 2005 International Conference on Information Quality, ICIQ 2005.

Radhakrishnan, A. and Das, S., 2018. Quality Assessment of Smart Grid Data. In: 2018 20th National Power Systems Conference, NPSC 2018.

Ramasamy, A. and Chowdhury, S., 2020. Big Data Quality Dimensions: a Systematic Literature Review. Journal of Information Systems and Technology Management, 17 (0).

Rismanchi, B., 2017. District energy network (DEN), current global status and future development. Renewable and Sustainable Energy Reviews [online], 75 (November 2016), 571–579. Available from: http://dx.doi.org/10.1016/j.rser.2016.11.025.

Rosinés, J. P., 2007. Jordi parés rosinés some on - line statistical methods for signal validation of redundant sensors 16.03.2007. University of Oulu.

Rusitschka, S. and Curry, E., 2016. Big data in the energy and transport sectors. New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe, 225–244.

Santhanam, T. and Padmavathi, M. S., 2014. Comparison of K-Means clustering and statistical outliers in reducing medical datasets. 2014 International Conference on Science Engineering and Management Research, ICSEMR 2014, 1–6.

Schneidewind, N., 2005. IEEE Standard For A Software Quality Metrics Methodology Revision And Reaffirmation, 1998, 278–278.

Schuelke-Leech, B. A., Barry, B., Muratori, M., and Yurkovich, B. J., 2015. Big Data issues and opportunities for electric utilities. Renewable and Sustainable Energy Reviews, 52, 937–947.

Sebastian-Coleman, L., 2013. Measuring Data Quality for Ongoing Improvement. Measuring Data Quality for Ongoing Improvement. Burlington: Morgan Kaufmann.

See, J., Carr, W., and Collier, S. E., 2008. Real time distribution analysis for electric utilities. Papers Presented at the Annual Conference - Rural Electric Power Conference, (08).

Sen, D., Aghazadeh, A., Mousavi, A., Nagarajaiah, S., Baraniuk, R., and Dabak, A., 2019. Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes. Mechanical Systems and Signal Processing [online], 131, 524–537. Available from: https://doi.org/10.1016/j.ymssp.2019.06.003.

Sha, K. and Shi, W., 2008. Consistency-driven data quality management of networked sensor systems. Journal of Parallel and Distributed Computing, 68 (9), 1207–1221.

Shafer, M. A., Fiebrich, C. A., Arndt, D. S., Fredrickson, S. E., and Hughes, T. W.,

2000. Quality assurance procedures in the Oklahoma Mesonetwork. Journal of Atmospheric and Oceanic Technology, 17 (4), 474–494.

Shalalfeh, L., Bogdan, P., and Jonckheere, E., 2020. Fractional Dynamics of PMU Data. IEEE Transactions on Smart Grid, 3053 (c).

Shekarpour, S. and Katebi, S. D., 2010. Modeling and evaluation of trust with an extension in semantic web. Journal of Web Semantics, 8 (1), 26–36.

Siddiqui, F., Sargent, P., and Montague, G., 2020. The use of PCA and signal processing techniques for processing time-based construction settlement data of road embankments. Advanced Engineering Informatics [online], 46 (July), 101181. Available from: https://doi.org/10.1016/j.aei.2020.101181.

Smith, D., Timms, G., De Souza, P., and D'Este, C., 2012. A Bayesian framework for the automated online assessment of sensor data quality. Sensors (Switzerland), 12 (7), 9476–9501.

Steinacker, R., Mayer, D., and Steiner, A., 2011. Data quality control based on self-consistency. Monthly Weather Review, 139 (12), 3974–3991.

Strong, D. M., Lee, Y. W., and Wang, R. Y., 1997. Data quality in context. Communications of the ACM, 40 (5), 103–110.

Taleb, I., Kassabi, H. T. El, Serhani, M. A., Dssouli, R., and Bouhaddioui, C., 2016. Big Data Quality : A Quality Dimensions Evaluation.

Tang, B., Gao, G., Xia, X., and Yang, X., 2018. Integrated energy system configuration optimization for multi-zone heat-supply network interaction. Energies, 11 (11).

Tayi, G. K. and Ballou, D. P., 1998. Examining Data Quality. Communications of the ACM, 41 (2), 54–57.

Timms, G. P., de Souza, P. A., Reznik, L., and Smith, D. V., 2011. Automated data quality assessment of marine sensors. Sensors, 11 (10), 9589–9602.

Timonen, J., 2018. Kaukolämmön Kysyntäjousto Kaukolämpötoimijoiden Näkö-Kulmasta, 99-117.

Tu, C., He, X., Shuai, Z., and Jiang, F., 2017. Big data issues in smart grid – A review. Renewable and Sustainable Energy Reviews, 79 (March), 1099–1107.

Vejen, F., Jacobsson, C., Fredriksson, U., Moe, M., Andresen, L., Hellsten, E., Rissanen, P., Palsdottir, T., and Arason, T., 2002. Quality Control of Meteorological Observations. Journal of Dementia Care.

Wand, Y. and Wang, R. Y., 1996. Anchoring data quality dimensions in ontological foundations. Communications of the ACM, 39 (11), 86–95.

Wang, R. Y. and Strong, D. M., 1996. Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems, 12 (4), 5–34.

Wang, R. Y., Ziad, M., and Lee, Y. W., 2001. Data quality. Boston (MA): Kluwer Academic Publishers.

Wang, X. and Makis, V., 2009. Autoregressive model-based gear shaft fault diagnosis using the Kolmogorov-Smirnov test. Journal of Sound and Vibration [online], 327 (3–5), 413–423. Available from: http://dx.doi.org/10.1016/j.jsv.2009.07.004.

Wienand, D. and Paulheim, H., 2014. Detecting incorrect numerical data in DBpedia. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8465 LNCS, 504–518.

Wolf, T., 2016. Meter Data Collection, Management, and Analysis. Smart Grid Handbook, (Mdm), 1–26.

Xiaojuan, B., Shurong, N., Zhaolin, X., and Peng, C., 2008. Novel method for the evaluation of data quality based on fuzzy control, 19 (3), 606–610.

Xie, S., Zheng, J., Hu, Z., Wang, J., and Chen, Y., 2020. Urban multi-energy network optimization: An enhanced model using a two-stage bound-tightening approach.

Applied Energy [online], 277 (July), 115577. Available from: https://doi.org/10.1016/j.apenergy.2020.115577.

Xu, X., Hu, Y., Tai, N., Fan, C., and Geng, Q., 2017. A reliable distribution network structure with the use of electric energy router. In: 2017 IEEE Conference on Energy Internet and Energy System Integration, EI2 2017 - Proceedings. 1–3.

Xu, Y., Zhang, J., Wang, W., Juneja, A., and Bhattacharya, S., 2011. Energy router: Architectures and functionalities toward energy internet. 2011 IEEE International Conference on Smart Grid Communications, SmartGridComm 2011, 31–36.

Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S., 2016. Quality assessment for Linked Data: A Survey. Semantic Web, 7 (1), 63–93.

Zhang, J., Ma, Y., and Hong, D., 2019. Research on Data Quality Assessment of Accuracy and Quality Control Strategy for Sensor Networks. Journal of Physics: Conference Series, 1288 (1).

Zhang, P., Xiong, F., Gao, J., and Wang, J., 2018. Data quality in big data processing: Issues, solutions and open problems. 2017 IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017 - , 1–7.

Zhou, K., Fu, C., and Yang, S., 2016. Big data driven smart energy management: From big data to big insights. Renewable and Sustainable Energy Reviews, 56 (2016), 215–225.

Zhou, K. and Yang, S., 2016. Understanding household energy consumption behavior: The contribution of energy big data analytics. Renewable and Sustainable Energy Reviews, 56, 810–819.

Zhou, K. and Yang, S., 2018. Smart Energy Management. Comprehensive Energy Systems.

Zikopolous, P. C., Eaton, C., DeRoos, D., Lapis, G., and Sit, S., 2012. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. Journal of Physics A: Mathematical and Theoretical.